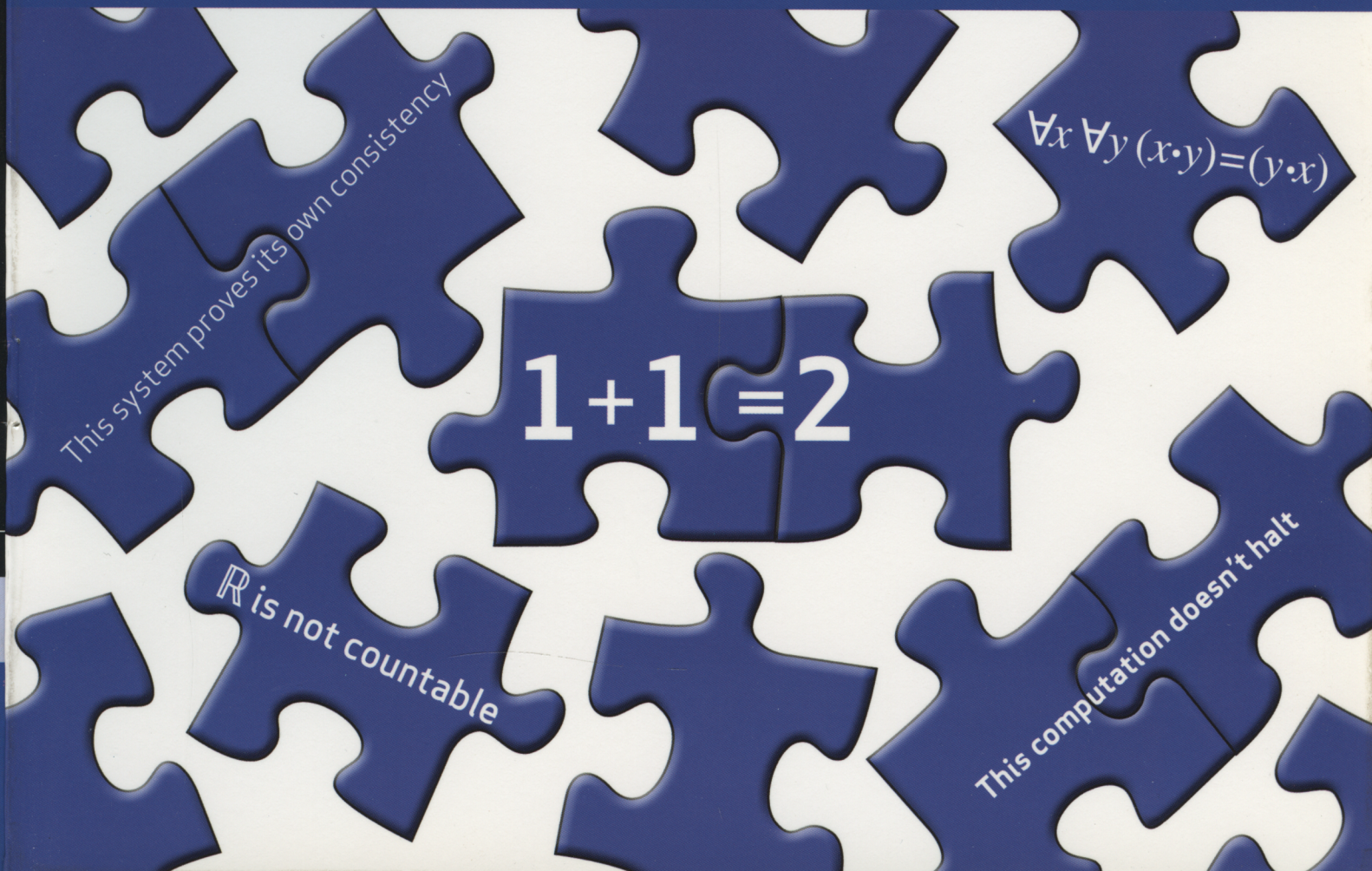
The Open University

M381 Number Theory and Mathematical Logic

Mathematical Logic **Unit 4**

Formal Systems



The Open University

M381 Number Theory and Mathematical Logic

Mathematical Logic **Unit 4**

Formal Systems

Prepared by the Course Team

The M381 Mathematical Logic Course Team

The Mathematical Logic half of the course was produced by the following team:

Roberta Cheriyan	*Course Manager*
Derek Goldrei	*Course Team Chair* and *Academic Editor*
Jeremy Gray	*History Consultant*
Mary Jones	*Critical Reader*
Roger Lowry	*Publishing Editor*
Alan Pears	*Author*
Alan Slomson	*Author*
Frances Williams	*Critical Reader*

with valuable assistance from:

The Maths Production Unit, Learning & Teaching Solutions:
Becky Browne, Jim Campbell, Nicky Kempton, Bill Norman, Sharon Powell, Katie Sayce, Penny Tee

Alison Cadle	*TEX Consultant*
Michael Goldrei	*Cover Design Consultant*
Vicki McCulloch	*Cover Designer*

The external assessor was:

Jeff Paris	*Professor of Pure Mathematics, University of Manchester*

The Course Team would like to acknowledge their reliance on the previous work of Alan Slomson and of Alex Wilkie, Professor of Mathematical Logic, University of Oxford.

This publication forms part of an Open University course. Details of this and other Open University courses can be obtained from the Student Registration and Enquiry Service, The Open University, PO Box 197, Milton Keynes, MK7 6BJ, United Kingdom: tel. +44 (0)870 300 6090, e-mail general-enquiries@open.ac.uk

Alternatively, you may visit the Open University website at http://www.open.ac.uk where you can learn more about the wide range of courses and packs offered at all levels by The Open University.

To purchase a selection of Open University course materials, visit http://www.ouw.co.uk, or contact Open University Worldwide, Michael Young Building, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom, for a brochure: tel. +44 (0)1908 858793, fax +44 (0)1908 858787, e-mail ouw-customer-services@open.ac.uk

The Open University, Walton Hall, Milton Keynes, MK7 6AA.

First published 2004. Reprinted as new edition 2007, with corrections.

Copyright © 2004 The Open University

All rights reserved; no part of this publication may be reproduced, stored in a retrieval system, transmitted or utilised in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without written permission from the publisher or a licence from the Copyright Licensing Agency Ltd. Details of such licences (for reprographic reproduction) may be obtained from the Copyright Licensing Agency Ltd, Saffron House, 6–10 Kirby Street, London EC1N 8TS; website http://www.cla.co.uk.

Open University course materials may also be made available in electronic formats for use by students of the University. All rights, including copyright and related rights and database rights, in electronic course materials and their contents are owned by or licensed to The Open University, or otherwise used by The Open University as permitted by applicable law.

In using electronic course materials and their contents you agree that your use will be solely for the purposes of following an Open University course of study or otherwise as licensed by The Open University or its assigns.

Except as permitted above you undertake not to copy, store in any medium (including electronic storage or use in a website), distribute, transmit or re-transmit, broadcast, modify or show in public such electronic materials in whole or in part without the prior written consent of The Open University or in accordance with the Copyright, Designs and Patents Act 1988.

Edited, designed and typeset by The Open University, using the Open University TEX System.

Printed in the United Kingdom by Charlesworth Press, Wakefield.

ISBN 978 0 7492 2267 3

3.1

CONTENTS

Introduction 4

1 The Formal Language 5

1.1 The basic symbols 6

1.2 Terms 9

2 Formulas 17

2.1 Formation rules for formulas 17

2.2 Gödel numbers 22

3 Interpretations 24

3.1 The interpretation of the logical connectives 25

3.2 Interpretation of the quantifiers and the arithmetical symbols 34

3.3 Logical consequences 39

Summary 41

Objectives 42

Additional Exercises 42

Solutions to the Problems 44

Solutions to Additional Exercises 54

Index 59

INTRODUCTION

Our work in the remainder of the course is directed towards Gödel's Incompleteness Theorems, which will be proved in *Unit 8*. For this purpose we must consider the idea of a *formal system*.

What is a formal system?

Mathematical statements are usually written in a natural language such as English, augmented with some special symbols such as $\frac{d}{dx}$ and $\int_a^b$. Natural languages have great advantages, but from our point of view they also have the defects of vagueness and ambiguity. In a *formal language*, however, we try to avoid these defects. We state explicitly what the basic symbols of the language are, these being the analogues of the letters of the alphabet and the punctuation marks of a natural language such as English. We also state precise rules for determining which strings of these basic symbols count as formulas of the language, these being the formal counterparts of meaningful statements in a natural language. This already contrasts with a natural language — could you give precise rules for deciding which strings of letters of the alphabet are English words? The rules for saying which strings count as formulas are called the *syntax* of the language. The way we give mathematical meaning to the formal language is called its *semantics*.

We use 'string' to mean a finite sequence of basic symbols, and 'formula' for those strings which comply with the rules of the language. As we shall see, the use of 'formula' here is rather broader than is usual in mathematics.

The intention is that these rules are such that there is an algorithm for determining which strings of our basic symbols are formulas of the language. In *Unit 2* we showed how we can assign code numbers to URM programs. Analogously we shall attach code numbers to strings of basic symbols. We saw that the characteristic function of the set of code numbers of URM programs is primitive recursive. It follows that there is a URM program which determines whether a given natural number codes a URM program. In a similar way it will turn out that there is a URM program which determines whether a given natural number codes a formula of our language. Thus the syntax of the language has an algorithmic character.

See Theorem 3.10 of *Unit 2*.

Usually, when carrying out mathematical arguments, we do not state explicitly which logical principles we are using. In a formal system the rules used in carrying out deductions are stated explicitly. These rules are applied to generate a sequence of formulas written in the formal language, and such a sequence, along with the explanation of how these rules have been applied to form it, constitutes a *formal proof*. Again, these rules have an algorithmic character, so we could devise a URM program which would check whether a purported formal proof really is a proof.

Thus a *formal system* consists of a formal language, including rules for determining the formulas of the language, together with rules for generating formal proofs.

What is the use of a formal system?

Recall that, in the Introduction to *Unit 1*, we asked you to keep in mind two questions at which this course is directed: Leibniz's Question and Hilbert's Question. Each of these questions provides a motivation for the setting up of a formal system for number theory.

Leibniz's Question: Is there an algorithm for deciding which statements of number theory are true?

In *Units 1* to *3* we have established what we mean by an algorithm and a decision procedure. To discover whether there is an algorithm for deciding which statements of number theory are true, we need first to be able to express the statements of number theory in a form to which algorithms may be applied, such as strings of symbols. Also, if we had no algorithm to decide which strings of symbols are formulas of the language, it would seem that we should have no hope of deciding which strings corresponded to *true* statements. Thus we must have a formal language for number theory in the sense described above.

Hilbert's Question: Can the consistency of number theory be proved using only non-dubious principles of finitary reasoning?

Hilbert wanted a completely clear and *finitary* proof that number theory is consistent: that is, a proof, involving only the mechanical manipulation of finite strings of symbols, that one could never obtain contradictory assertions about natural numbers. In order to make sense of this question, clearly we require a precise notion of what a proof is. Once we have formulated this, then formal proofs themselves become the objects of mathematical investigation, just as numbers are the objects of study in number theory. Indeed, just as the natural numbers of number theory are finite objects and are manipulated according to mechanical rules, formal proofs are also finite objects and, in a formal system, they are manipulated using mechanical rules. This gives rise to the hope that we may be able to show, using simple reasoning, that these manipulations cannot lead to a contradiction. That is, we can try to carry out Hilbert's programme and aim to give a finitary consistency proof for number theory.

These then are the issues that we shall explore in this unit and in *Units 5* to *8*. In this unit we shall introduce the formal language with which we shall express statements about number theory. We shall see how to represent formal statements by numbers, called *Gödel numbers*, to which algorithms of the sort we have seen in *Units 1* to *3* can be applied. We shall look at the way in which formal statements are interpreted so that they can be seen as representing statements about $\mathbb{N}$ and about other sets with operations on them. And finally we shall begin to look at what it means for one statement to be a consequence of others, to give a yardstick for what we expect from the formal proof system. In *Units 5* and *6* we shall develop the proof system for handling formal statements and we shall investigate a set of axioms for number theory. In *Unit 7* we shall see that, with these axioms, the proof system has the power to represent recursive functions. In *Unit 8* we shall see how Gödel knitted all these ideas together to answer Leibniz's and Hilbert's Questions.

1 THE FORMAL LANGUAGE

In this section we shall look at the *basic symbols* from which we can construct the formal statements, or formulas, that we shall use to represent statements about number theory. We shall also look at those combinations of the symbols, called *terms*, which will represent numbers and arithmetic expressions. We shall also discuss an algorithm for distinguishing between those combinations of symbols which are terms and those which are not.

1.1 The basic symbols

In devising a formal system for number theory we are pulled in different directions by two opposing considerations. We want the system to be sufficiently rich so that within it we can state and prove significant theorems of number theory. Without this it would hardly count as a formal system for number theory. On the other hand our intention is that the formulas and proofs of our formal system should themselves be the objects of mathematical study, and to this end it is desirable to keep things as simple as possible.

We aim to steer a path between these extremes, but in formulating the language and the rules for formal proofs our bias will be towards simplicity. If we listed all the symbols used, for example, in the Number Theory units, we would end up with a fairly long list. However, it turns out that we can manage with remarkably few basic symbols.

Consider, for example, how we might use symbols to state Euclid's Theorem:

Number Theory, *Unit 2*, Theorem 3.1.

> There are infinitely many primes.

We would need to find a way to express, in symbols, the property of being a prime number and the statement that there are infinitely many numbers with this property.

A convenient way to express the fact that there are infinitely many primes is to say that there is no largest prime number. We can do this by saying that, given any number, there is always a prime number which is at least as large as the given number. If we use $\forall$ to mean 'for all', $\exists$ for 'there exists', and the symbol & to mean 'and', we can write Euclid's Theorem as

$$\forall x \, \exists y \, (\text{'}y \text{ is prime'} \ \& \ x \leq y)$$

We next need to replace the English phrase 'y is prime' by symbols. We know that a number y is prime if $y \neq 1$ and the only divisors of y are 1 and y itself. So if we use $z \mid y$ for 'z divides y' then we can express 'y is prime' by

The notation $z \mid y$ was introduced in Number Theory, *Unit 1*.

$$(y \neq 1 \ \& \ \forall z \, (z \mid y \rightarrow (z = 1 \text{ or } z = y))) \tag{1.1}$$

Here we have used the symbol $\rightarrow$ to mean 'implies' (that is 'if ... then ...'). We ought to say here that we are regarding all our variables (x, y, z and so on) as referring to natural numbers, that is to 0, 1, 2, 3, ..., rather than to, say, integers or real numbers. Thus the first part of (1.1), before the & symbol, says that 'y is not equal to 1' and the second part, after the & symbol, says that 'for all natural numbers z, if z divides y then z is equal to 1 or z is equal to y'.

Thus we have the following symbolic expression for Euclid's Theorem:

$$\forall x \, \exists y \, ((y \neq 1 \ \& \ \forall z \, (z \mid y \rightarrow (z = 1 \text{ or } z = y))) \ \& \ x \leq y)$$

However, we can simplify things even more. We are almost certainly going to need symbols for the basic arithmetical operations of addition and multiplication, so it is worth noting that divisibility can be defined in terms of multiplication and the $\leq$ relation in terms of addition. Thus we can replace $z \mid y$ by $\exists t \, (z \cdot t = y)$ which says that there is a natural number t such that $z \times t = y$, that is, there is a multiple of z which equals y. Also $x \leq y$ can be replaced by $\exists s \, (x + s = y)$.

The final simplification we make at this stage is to use the symbol $\vee$ in place of the word 'or'. Putting all this together we see that Euclid's Theorem can be expressed as

$$\forall x \, \exists y \ ((y \neq 1 \ \& \ \forall z \, (\exists t \, (z \cdot t = y) \rightarrow (z = 1 \vee z = y))) \ \& \ \exists s \, (x + s = y))$$

Thus Euclid's Theorem can be expressed using a limited stock of symbols. Indeed the symbols we have used to do this are almost all we need for a language powerful enough for number theory.

Some of the symbols we have used, for example, the symbols &, $\vee$, $\rightarrow$, $\forall$, $\exists$ and $=$ would be needed in any mathematical theory. These are called *logical symbols*, and we shall soon introduce more. These logical symbols include symbols for *variables*, which are needed by any mathematical theory to refer to the objects under consideration in that theory. Since we shall be interested in number theory, we intend that the variables should refer to natural numbers, though the logical rules that we shall introduce will be valid no matter to which set of objects the variables refer.

Other symbols, such as $+$ and $\cdot$, are very special to number theory. We shall call them *arithmetical symbols*. The arithmetical symbols include symbols for addition and multiplication, and apparatus to provide symbols for all the natural numbers. You might think that this would involve using ten digits. We can, however, make do with a symbol $\mathbf{0}$ to denote zero and a symbol $'$ to denote the operation 'add one'. Thus, for example, we can denote the number two by $\mathbf{0''}$. Because we shall be interested in number theory, we intend the arithmetical symbols $+$ and $\cdot$ to represent the usual arithmetical operations of addition and multiplication on the set of natural numbers. However, the formal system we describe will handle these symbols in a purely mechanical way that does not take into account their intended interpretation. To distinguish these symbols from their interpretations, we shall use a heavier print for the symbols, for example $+$ rather than $+$ and $\cdot$ rather than $\cdot$.

The distinction between a symbol and its interpretation is important for the study of the subject. But there is no need for you to try to struggle to make this distinction when reading formulas in the units or writing them down in your assessment answers. The context will usually make it clear whether the symbol or its interpretation is meant.

There is no limit to the number of variables that we may need. We used five variables, s, t, x, y, z, in expressing Euclid's Theorem above. More complicated theorems will need even more variables. Since it is a key feature of our approach that our formal language is built up from a finite set of symbols, we provide for an unlimited number of symbols by using just two symbols x and $_1$ from which we can generate the unending list

$$x,\ x_1,\ x_{11},\ x_{111},\ \ldots$$

of strings of symbols to serve as variables. For clarity and brevity we shall usually write these as

$$x_0,\ x_1,\ x_2,\ x_3,\ \ldots.$$

We are now ready to specify our basic symbols.

Definition 1.1 Basic symbols

The *basic symbols* of the formal language are:

(a) *Logical symbols*: These come in five categories.

(i) *Connectives*: $\neg$ & $\vee$ $\rightarrow$ $\leftrightarrow$
(called, respectively, negation, conjunction, disjunction, implication and bi-implication).

(ii) *Quantifiers*: $\forall$ $\exists$
(called, respectively, the universal quantifier and the existential quantifier).

(iii) *Identity symbol*: $=$

(iv) *Punctuation symbols*: ( )

(v) *Variables*: x_0 x_1 x_2 ...

(b) *Arithmetical symbols*: $\mathbf{0}$ $'$ $+$ $\cdot$
(known, respectively, as the zero symbol, the successor symbol, the addition symbol and the multiplication symbol).

The connectives are sometimes referred to as *logical connectives*.

We have already mentioned the intended meanings for the connectives & ('and'), $\vee$ ('or') and $\rightarrow$ ('implies' or equivalently 'if ... then ...') and for the quantifiers $\forall$ ('for all') and $\exists$ ('there is' or 'there exists'). It might help you

We shall discuss the interpretation of all the symbols more fully in Section 3.

to know that the intended meaning of the connective $\neg$ is 'not' and of $\leftrightarrow$ is 'if and only if'. Also note that, as you no doubt expect, the intended meaning of the identity symbol $=$ is to express the relation of equality.

Before giving the precise rules for deciding which strings of the symbols we have introduced are to be regarded as formulas of our formal language, let us take some examples of informal statements of number theory and formulate them using only the basic symbols. We hope that this will give you some idea of how mathematically rich these few symbols are.

One example is Euclid's Theorem, which we have already seen can be written as

$$\forall x\, \exists y\, ((y \neq 1 \,\&\, \forall z\, (\exists t\, (z \cdot t = y) \rightarrow (z = 1 \vee z = y))) \,\&\, \exists s\, (x + s = y))$$

To express this in terms of the basic symbols we need to replace $y \neq 1$ by $\neg\, y = 1$ the variables x, y, z, t, s by x_0, x_1, x_2, x_3, x_4, say, and 1 by $\mathbf{0'}$, to obtain

$$\forall x_0\, \exists x_1\, ((\neg\, x_1 = \mathbf{0'} \,\&\, \forall x_2\, (\exists x_3\, (x_2 \cdot x_3 = x_1) \rightarrow (x_2 = \mathbf{0'} \vee x_2 = x_1))) \,\&\, \exists x_4\, (x_0 + x_4 = x_1))$$

You will see later that this expression for Euclid's Theorem, using basic symbols, needs to be adjusted slightly to turn it into a correct formula of our formal language.

Example 1.1

Every number is either even or odd.

Firstly we note that 'x_0 is even' is the same as 'x_0 is a multiple of 2'. As 2 can be represented in our language by $\mathbf{0''}$, this can be written in terms of the basic symbols as

$$\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1)$$

Similarly 'x_0 is odd' is the same as 'x_0 is of the form $2x + 1$' and, representing 1 by $\mathbf{0'}$, this can be represented in terms of basic symbols by

$$\exists x_1\, x_0 = ((\mathbf{0''} \cdot x_1) + \mathbf{0'})$$

Thus 'every number is either even or odd' can be written as

$$\forall x_0\, (\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1) \vee \exists x_1\, x_0 = ((\mathbf{0''} \cdot x_1) + \mathbf{0'})) \qquad (1.2)$$

which is just a string of basic symbols. ♦

One thing to note about (1.2) in Example 1.1 is that it isn't the only possible formulation. For example, we could have written

$$\forall x_0\, (\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1) \vee \exists x_2\, x_0 = ((\mathbf{0''} \cdot x_2) + \mathbf{0'})) \qquad (1.3)$$

and you may indeed find this formulation clearer — the use of different variables x_1 and x_2 either side of the $\vee$ connective makes the expression easier to interpret.

There is further discussion of the use of variables in *Unit 5*.

Example 1.1 also illustrates the importance of *brackets* — what we have described in Definition 1.1 as punctuation symbols. Without brackets we wouldn't know whether the string

$$\mathbf{0''} \cdot x_1 + \mathbf{0'}$$

was meant to stand for $(2 \times x_1) + 1$ or $2 \times (x_1 + 1)$. Given that we want it to stand for the first of these, we need to insert at least one pair of brackets to make it read as

$$(\mathbf{0''} \cdot x_1) + \mathbf{0'}$$

But, in case we might later want to incorporate this in a more complicated expression, we give it an outer pair of brackets to obtain

$$((\mathbf{0''} \cdot x_1) + \mathbf{0'})$$

Similarly we have used pairs of brackets to help us work out how the statements 'x_0 is even' and 'x_0 is odd' are to be joined up and that the 'for all x_0' covers both, rather than one, of them. For instance, by placing brackets differently we can obtain

$$(\forall x_0\, \exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1) \vee \exists x_1\, x_0 = ((\mathbf{0}'' \cdot x_1) + \mathbf{0}'))$$

which could be read as 'every number is even or x_0 is odd'.

The message is that brackets are going to be very important!

Example 1.2

Every number has a square root.

Your first reaction might be 'But this is false!' as we are dealing just with the natural numbers. However, the truth or falsity of these statements is not our present concern. For the moment we just wish to show that this informal statement can be expressed by a string of basic symbols. Indeed, if we are to tackle Leibniz's Question, we shall need to be able to express statements of number theory without knowing whether they are true or false.

We do not need to introduce a square root symbol in order to express 'x_0 has a square root' because this can be rewritten as 'x_0 is the square of some number' and hence may be written as $\exists x_1\, x_1^2 = x_0$, and x_1^2 may be written in terms of basic symbols as $(x_1 \cdot x_1)$. Thus we may write 'x_0 has a square root' as $\exists x_1\, (x_1 \cdot x_1) = x_0$. Hence the informal statement 'every number has a square root' can be written as

$$\forall x_0\, \exists x_1\, (x_1 \cdot x_1) = x_0$$

using only basic symbols. ♦

Problem 1.1

Formulate the statement 'the square of every even number is even' using basic symbols.

1.2 Terms

As was mentioned in the Introduction, our formal language is made up of strings of the basic symbols called *formulas*. The strings of basic symbols which we ended up with in Examples 1.1 and 1.2 are examples of formulas. The most simple formulas are strings which consist of equations between numerical expressions (such as $x_0 = ((\mathbf{0}'' \cdot x_1) + \mathbf{0}')$ in Example 1.1 and $(x_1 \cdot x_1) = x_0$ in Example 1.2). More complicated formulas are built up from these equations using the connectives and the quantifiers. So, to begin the process of specifying rules for creating formulas, we must specify exactly which strings can occur on either side of an identity symbol to form an equation. We call strings of this sort *terms*, and they form the subject-matter of this subsection.

The rules for creating formulas from terms are discussed in Section 2.

Roughly speaking, terms are built up from the constant $\mathbf{0}$ and variables, using the symbols $'\ +\ \cdot$ in an appropriate way. We can see how to turn this rough description into a precise specification if we think, for example, about how we could justify the claim that the string of basic symbols

$$((\mathbf{0}'' \cdot x_1) + \mathbf{0}')$$

is a term. In full detail the argument would go as follows.

(1) $\mathbf{0}$ is a term (constant)

(2) $\mathbf{0}'$ is a term ($'$ applied to (1))

(3) $\mathbf{0}''$ is a term ($'$ applied to (2))

(4) x_1 is a term (variable)

(5) $(\mathbf{0}'' \cdot x_1)$ is a term ($\cdot$ applied to (3) and (4))

(6) $((\mathbf{0}'' \cdot x_1) + \mathbf{0}')$ is a term ($+$ applied to (5) and (2))

Note that an application of $+$ or $\cdot$ is always accompanied by the inclusion of a pair of outer brackets.

It can be seen that this argument consists of a finite sequence of terms culminating in the string in which we are interested, together with a justification of why each string in the sequence is a term. In each case the justification is one of three types:

(a) the given string is either the constant $\mathbf{0}$ or a variable;

(b) the given string has been obtained from an earlier string using $'$;

(c) the given string has been obtained from a pair of earlier strings using $+$ or $\cdot$, with outer brackets (and) added.

There is no real need to add these justifications as it is straightforward to check just by looking at the sequence of strings whether each line is justified in one or other of these ways. So the above argument could be replaced by the following sequence:

$$\mathbf{0},\ \mathbf{0}',\ \mathbf{0}'',\ x_1,\ (\mathbf{0}'' \cdot x_1),\ ((\mathbf{0}'' \cdot x_1) + \mathbf{0}').$$

Note that in a sequence of strings of this type, which complies with conditions (a), (b) and (c), the order is important, but usually more than one ordering is possible. For example in the above sequence it is important that $\mathbf{0}$ occurs before $\mathbf{0}'$ and that $\mathbf{0}'$ occurs before $\mathbf{0}''$. Also, both $\mathbf{0}''$ and x_1 must occur before $(\mathbf{0}'' \cdot x_1)$. However, the position of x_1 as the fourth string in the sequence is not crucial. It could occur in any of the first four positions.

As it stands, we have not eliminated the possibility of including redundant strings in a sequence of the above type. Thus, for example, the sequence

$$\mathbf{0},\ \mathbf{0}',\ \mathbf{0}'',\ x_1,\ x_3,\ x_3',\ x_3'',\ x_4,\ (x_3'' + x_4),\ (\mathbf{0}'' \cdot x_1),\ ((\mathbf{0}'' \cdot x_1) + \mathbf{0}')$$

also satisfies conditions (a), (b) and (c) and thus justifies the claim that the final string is a term. It can be seen that the strings x_3, x_3', x_3'', x_4, $(x_3'' + x_4)$ have been inserted into the earlier sequence, and that these strings are redundant as they do not occur anywhere in the final string. To avoid this sort of redundancy, we shall add the requirement that each string in a sequence of this kind is a substring of the final string, where we say that the string s is a *substring* of the string t if s is made up of consecutive symbols from t. Thus, for example, the substrings of the string $(\mathbf{0} + \mathbf{0}')$ include (, $(\mathbf{0},\ \mathbf{0} + \mathbf{0}',\ \mathbf{0}'$ and $+\ \mathbf{0}')$.

There are some sorts of redundancy which we won't try to avoid. You will encounter an example in Problem 1.2.

We can now give a precise specification of which strings are terms.

Definition 1.2 Term

A string τ of basic symbols is a *term* if and only if there is a finite sequence of strings $\tau_1, \tau_2, \ldots, \tau_m$ of basic symbols such that, for $i < m$, each τ_i is a substring of τ, the string τ_m is τ, and for $1 \leq i \leq m$ one of the following holds:

(a) τ_i is the constant $\mathbf{0}$ or a variable;

(b) for some $j < i$, τ_i is τ_j';

(c) for some $j, k < i$, τ_i is $(\tau_j + \tau_k)$ or $(\tau_j \cdot \tau_k)$.

τ is the Greek letter 'tau'.

Observe that we need a pair of outer brackets for each occurrence of either $+$ or $\cdot$ in a term.

Example 1.3

$$((x_0 \cdot x_2) + (x_2 + \mathbf{0}'))$$

is a term. This follows from the fact that the sequence

$$\mathbf{0},\ x_0,\ x_2,\ \mathbf{0}',\ (x_0 \cdot x_2),\ (x_2 + \mathbf{0}'),\ ((x_0 \cdot x_2) + (x_2 + \mathbf{0}'))$$

complies with the conditions of Definition 1.2. ♦

Note that every string in the sequence in Example 1.3 is a term. Indeed if the sequence of strings $\tau_1, \tau_2, \ldots, \tau_m$ satisfies the conditions of Definition 1.2, then for each $i < m$ so also does the sequence $\tau_1, \tau_2, \ldots, \tau_i$, so that every string in the sequence $\tau_1, \tau_2, \ldots \tau_m$ is a term. The sequence shows us how the final term is built up from earlier terms in the sequence.

Example 1.4

(a) The sequence

$$x_0,\ \mathbf{0}',\ (x_0\mathbf{0}')$$

does not comply with the requirements of Definition 1.2. Indeed, the string $(x_0\mathbf{0}')$ is not a term — any term involving the terms x_0 and $\mathbf{0}'$ as substrings would also have to involve $+$ or $\cdot$ to join them together.

(b) The sequence

$$\mathbf{0},\ x_2,\ x_3,\ (\mathbf{0} \cdot x_2),\ x_3 + (\mathbf{0} \cdot x_2)$$

does not comply with the requirements of Definition 1.2. The string $x_3 + (\mathbf{0} \cdot x_2)$ is not a term as it has insufficient brackets. Each $+$ or $\cdot$ appearing in a term requires a corresponding pair of outer brackets. ♦

Correct use of brackets plays a very important part in the formation of terms.

Example 1.5

Neither of the sequences

(a) $\mathbf{0},\ \mathbf{0}',\ (\mathbf{0}' + x_2),\ x_1,\ x_2,\ (x_1 \cdot (\mathbf{0}' + x_2))$

(b) $\mathbf{0},\ \mathbf{0}',\ x_1,\ (x_1 + \mathbf{0}'),\ (x_2' + (x_1 + \mathbf{0}'))$

complies with the requirements of Definition 1.2, but in each case the final string is a term. We can modify each sequence so that it does meet the requirements of the definition and thus shows that the final string is a term. In (a) all that is necessary is to move the string x_2 so that it precedes the string $(\mathbf{0}' + x_2)$. In (b) we need to insert the strings x_2, x_2' in this order anywhere in the sequence before the final string. ♦

Problem 1.2

(a) Which of the following sequences complies with the requirements of Definition 1.2?

(i) $\mathbf{0},\ \mathbf{0}',\ (\mathbf{0} + \mathbf{0}'),\ x_2,\ (x_2 \cdot (\mathbf{0} + \mathbf{0}'))$

(ii) $x_1,\ x_2,\ (x_1 + \mathbf{0}''),\ (x_2 \cdot (x_1 + \mathbf{0}''))$

(iii) $\mathbf{0},\ \mathbf{0}',\ x_1,\ (x_1 \cdot \mathbf{0}'),\ (x_1 \cdot \mathbf{0}')',\ x_1,\ ((x_1 \cdot \mathbf{0}')' + x_1)$

(b) In which of the above cases is the final string a term? Justify your answer.

Definition 1.2 gives us a precise characterization of which strings are terms, but it is not yet evident that it provides us with an algorithm for determining whether a given string is a term. If a string is a term, how do we find the sequence which the definition requires to justify that it is a term? For a short string this may not be too difficult, but for a very long string we need a systematic method. Furthermore, how do we show that a string is not a term? We have seen from Example 1.5 that the fact that a sequence of strings does not comply with the definition does not prove that the final string is not a term. Again, when a string is short, as in

Example 1.4, it may be immediately evident that it fails to be a term, but we seek a method which will work for all strings however long. We now turn our attention to this issue.

We begin with an observation which is of theoretical interest but which does not lead to a very practical method for determining which strings are terms. When we have a sequence $\tau_1, \tau_2, \ldots, \tau_m$ of strings which complies with Definition 1.2, and which hence shows that the final string, τ_m, is a term, all the earlier strings are substrings of the final string. This means that the number of strings which can occur in such a sequence is strictly limited and we can systematically list all the possibilities. Again, a given set of these substrings can be arranged in order in only a finite number of ways. So we can systematically list all possible arrangements of the subsets of the set of substrings of a string τ and test each of them to see whether it complies with Definition 1.2. If we find one which does, we can deduce that τ is a term. If none of them meets the requirements of the definition, then τ is not a term.

Although this gives us a theoretical algorithm for determining which strings are terms, it is not of much practical use because, for all but very short strings, the number of different sequences made up of substrings of the given string (not allowing repetitions) is very large.

Fortunately, there is a more practical algorithm, which we now explain.

First note that a string consisting of more than one symbol can only be a term if it contains at least one occurrence of one of $+$, $\cdot$ and $'$. If such a string ends with a $'$, then we remove it and apply our algorithm to the shorter string. Note that a string of the form τ' is a term if and only if the string τ is a term. If a string with more than one symbol is to be a term and doesn't end with a $'$, it has to be of the form $(\tau_1 + \tau_2)$ or $(\tau_1 \cdot \tau_2)$, so it has to start with a left-hand bracket '(' and end with a right-hand bracket ')'.

What we do next is illustrated by showing how it works with the string

$$(((x_1' \cdot x_2'') + \mathbf{0''}) \cdot (x_1 + (\mathbf{0''} \cdot x_2)))$$

It follows from clause (c) of Definition 1.2 that, in a term, the number of left-hand brackets equals the number of right-hand brackets. We can check this for a given string in the following way. Starting at the left-hand end of the string with the number 0, we add 1 for each left-hand bracket and subtract 1 for each right-hand bracket. In this way we associate a number with each bracket. For the string above this gives

$$\begin{array}{llllllllllllllllll} (& (& (& (& x_1' & \cdot & x_2'' &) & + & \mathbf{0''} &) & \cdot & (& x_1 & + & (& \mathbf{0''} & \cdot & x_2 &) &) &) \\ 1 & 2 & 3 & & & & 2 & & & 1 & 2 & & 3 & & & & 2 & 1 & 0 & & & \end{array}$$

The fact that the final bracket in the string is associated with the number 0 tells us that the string contains equal numbers of left-hand and right-hand brackets.

This bracket-counting procedure is useful in another way, as we shall see below. In the discussion, the only symbols we need consider are the arithmetical symbols $+$ $\cdot$ and the brackets ( ). It will help us to have a collective name for these symbols, so we shall call them *a-symbols*.

There is no standard name for what we have called an a-symbol.

We see that, in the example above, there are three brackets which are associated with the number 1. In just one of these cases, the next a-symbol after the bracket is an arithmetical symbol, whereas in the other two cases it is another bracket. This arithmetical symbol, associated with exactly one of the brackets numbered 1, will be called the *principal symbol* of this string.

It can be proved that a term not ending in a $'$ and containing one or more occurrences of the arithmetical symbols $+$ $\cdot$ always contains *just one* principal symbol.

The whole string has the form $(\tau_1 \cdot \tau_2)$ with the principal symbol between the strings τ_1 and τ_2:

$$(\underbrace{\boxed{((x_1' \cdot x_2'') + \mathbf{0}'')}}_{\tau_1} \cdot \underbrace{\boxed{(x_1 + (\mathbf{0}'' \cdot x_2))}}_{\tau_2})$$

It follows from Definition 1.2 that if the strings τ_1 and τ_2 are both terms, then so also is the string $(\tau_1 \cdot \tau_2)$. It is also true for a string of the form $(\tau_1 \cdot \tau_2)$ that if either of τ_1 and τ_2 fails to be a term, then the whole string is not a term. So, using this analysis, we have reduced the question as to whether the original string is a term to the questions as to whether the shorter strings τ_1 and τ_2 are terms. We can answer these questions by carrying out the same analysis with the strings τ_1 and τ_2.

The bracket counts for these strings are as follows

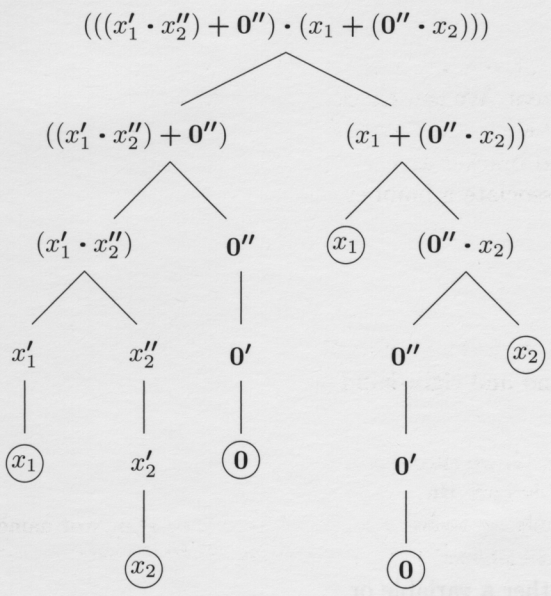
$$\begin{array}{llll} \tau_1 & ((x_1' \cdot x_2'') + \mathbf{0}'') & \tau_2 & (x_1 + (\mathbf{0}'' \cdot x_2)) \\ & 12 \quad 1 \quad 0 & & 1 \quad 2 \quad 10 \end{array}$$

It can be seen that in both cases the string doesn't end with a $'$, starts and ends with the correct sorts of bracket, and has the arithmetical symbol $+$ as its principal symbol. We use this to split up both τ_1 and τ_2, just as we did with the original string.

In this process the strings we are analysing become shorter and shorter, and so the process must halt with strings which cannot be analysed any further. If all the strings we end up with in this way are either variables or the constant $\mathbf{0}$, then the original string is a term; if not, then it is not a term.

Fortunately this process of analysis usually takes up less space to carry out than to describe. So we now illustrate it with several examples, beginning with the string we have used above. The analysis of this string is set out in the following diagram.

Note that, although the bracket-counting procedure is used to create the diagram, we would normally omit the numbers used in that procedure from the diagram.



Each branch of this diagram ends with a term which is either a variable or the constant $\mathbf{0}$. Circles have been put around these terms. Since each branch ends with a variable or the constant $\mathbf{0}$, we can deduce that the string at the top is a term. Every string which is a term can be analysed in this way.

If we read the diagram upwards, we see how the term at the top is built up from variables and the constant $\mathbf{0}$ using the arithmetical symbols $+ \cdot '$. We can turn the diagram into a finite sequence of terms, as required by Definition 1.2, very easily. The required sequence must include all the terms which occur in the diagram. We must ensure that each term in the sequence is preceded by the term or terms immediately below it. Terms that occur more than once in the diagram (for example, $\mathbf{0}$, $\mathbf{0}'$, $\mathbf{0}''$, x_1 and x_2) do not need to be repeated in the sequence, provided that the condition of the previous sentence is satisfied. Here is one example of a sequence derived from the diagram which meets this last condition and also satisfies the conditions of Definition 1.2:

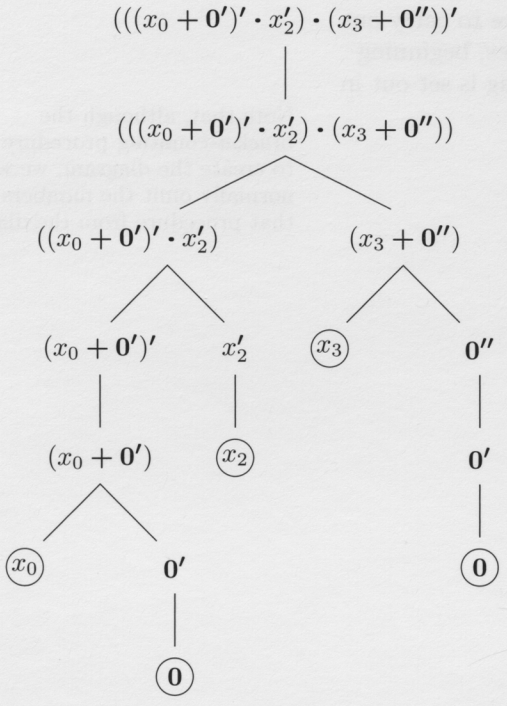
$$\mathbf{0},\ x_1,\ x_2,\ \mathbf{0}',\ \mathbf{0}'',\ x_1',\ x_2',\ x_2'',\ (x_1' \cdot x_2''),\ (\mathbf{0}'' \cdot x_2),\ ((x_1' \cdot x_2'') + \mathbf{0}''),$$
$$(x_1 + (\mathbf{0}'' \cdot x_2)),\ (((x_1' \cdot x_2'') + \mathbf{0}'') \cdot (x_1 + (\mathbf{0}'' \cdot x_2)))$$

Example 1.6

We analyse the string

$$(((x_0 + \mathbf{0}')' \cdot x_2') \cdot (x_3 + \mathbf{0}''))'$$

Since this string ends with the symbol $'$, the first step in the analysis is to consider the string obtained by removing this final symbol. This latter string begins and ends with the correct sorts of bracket. The bracket count then reveals that the principal symbol of the remaining string is the second $\cdot$ (counting, as usual, from left to right). Then the analysis proceeds as shown in the following diagram.



Each branch of this diagram ends with a term which is either a variable or the constant $\mathbf{0}$. We therefore deduce that the original string is a term. ♦

Problem 1.3

Give a finite sequence of terms which complies with the requirements of Definition 1.2 and which shows that the string of Example 1.6 is a term.

Example 1.7

Consider the string

$$+\mathbf{0}(x_0)\cdot$$

It should be immediately evident that this string is not a term. This is confirmed by our analytical method: it consists of more than one symbol, it doesn't end in a $'$ and it doesn't start with a left-hand bracket '(', so it is not a term. ♦

Example 1.8

Consider the string

$$((x_3 + (x_1 \cdot \mathbf{0}'))) \cdot x_2'$$

The bracket count

$$\begin{array}{llllll} (& (& (x_3 + (& x_1 \cdot \mathbf{0}' &) &)) \cdot x_2' \ ) \\ 1 & 2 & 3 & 2 1 0 & & -1 \end{array}$$

immediately reveals that the number of left-hand brackets is not equal to the number of right-hand brackets. Hence this string is not a term. ♦

Example 1.9

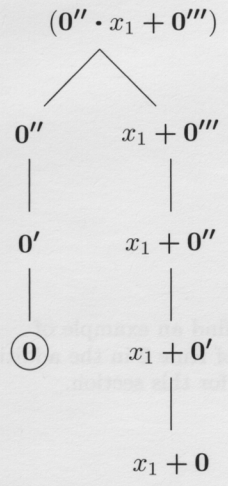
Consider the string

$$(\mathbf{0}'' \cdot x_1 + \mathbf{0}''')$$

The bracket count

$$\begin{array}{lll} (& \mathbf{0}'' \cdot x_1 + \mathbf{0}''' &) \\ 1 & & 0 \end{array}$$

reveals that the symbol $\cdot$ is the principal symbol. We then have the following analysis:



The left-hand branch of the diagram ends with the constant $\mathbf{0}$. The right-hand branch begins with the string $x_1 + \mathbf{0}'''$. As this string ends with a $'$, we remove this and analyse the string $x_1 + \mathbf{0}''$. As this string ends with two $'$s, we repeat this process twice more to obtain the string $x_1 + \mathbf{0}$. As this consists of more than one symbol, doesn't end with a $'$ and has no outer brackets, it is not a term. Hence the original string $(\mathbf{0}'' \cdot x_1 + \mathbf{0}''')$ is not a term. ♦

We now list explicitly the rules that we use in carrying out an analysis of a string to see if it is a term.

1 If the string is of the form τ', we write under it the string τ.

2 If the first symbol is a left-hand bracket, the last symbol is a right-hand bracket and the string includes at least one of the arithmetical symbols $+$ $\cdot$, then we perform a bracket count. If this shows that there are equal numbers of left-hand and right-hand brackets, and there is just one bracket which is associated with the number 1 and is such that the next a-symbol following this bracket is $+$ or $\cdot$, then this a-symbol is the principal symbol of the string. In this case the string has the form $(\sigma + \tau)$ or $(\sigma \cdot \tau)$, where $+$ or $\cdot$ is the principal symbol, and we write the pair of strings σ, τ below it.

   The bracket count may reveal in one of two ways that the string we are considering is not a term.

   (a) It may show that there are unequal numbers of left-hand and right-hand brackets.

   (b) It may show that there is no principal symbol, either because there is no bracket associated with the number 1 such that the next a-symbol is $+$ or $\cdot$, or because there is more than one such bracket.

   In either of these cases we describe the string as a *terminal string* and our analysis can terminate. The string we are considering is not a term, and it follows that the initial string is not a term.

3 If neither of Rules 1 and 2 applies, we describe the string as a *terminal string.*

All branches of the analysis must end with terminal strings. It can be shown that the original string is a term if and only if each of the terminal strings has the form $\mathbf{0}$ or v, where v is a variable.

You have seen several examples of the use of Rule 2. For example, in Example 1.6, we applied it to the string

$$((x_0 + \mathbf{0}')' \cdot x_2')$$

The bracket count for this string is

$$\begin{array}{ccccc} (& (& x_0 + \mathbf{0}')' & \cdot & x_2') \\ 1 & 2 & & 1 & 0 \end{array}$$

This shows that there are equal numbers of left-hand and right-hand brackets, and that $\cdot$ is the principal symbol. Thus the string is of the form $(\sigma \cdot \tau)$ where σ is the string $(x_0 + \mathbf{0}')'$ and τ is the string x_2', and our analysis proceeds by analysing the two strings σ and τ. Example 1.8 provides an example of case (a) of Rule 2.

You will find an example of case (b) of Rule 2 in the additional exercises for this section.

Problem 1.4

Analyse each of the following strings of basic symbols, and hence decide which are terms and which are not.

(a) $(((\mathbf{0} \cdot x_1) \cdot x_1)'' + (x_2 + \mathbf{0}''))$

(b) $(((\mathbf{0} + x_1) + x_2) + x_1 \cdot x_2)$

(c) $(x_6'' \cdot (x_9 + \mathbf{0}'''))'$

(d) $(((x_1 \cdot \mathbf{0}') \cdot \forall x_2) + (x_3'' \cdot \mathbf{0}))$

It should be reasonably clear from the discussion and examples in this subsection that our method for analysing strings provides an algorithm for determining whether a string is a term.

This is made more precise in Subsection 2.2.

Although all the discussion in this subsection has been devoted to the definition of terms, the approach we have used has a much wider application, as we shall see in the next section where we discuss the definition of *formulas* in our formal language.

2 FORMULAS

In this section we shall complete our description of the formal language by explaining how to construct formal statements, or *formulas*, involving the terms discussed in the previous section. We shall also discuss an algorithm for distinguishing between those combinations of symbols which represent formulas and those which do not — this algorithm is very similar to the one for terms described in Section 1. The section ends with a discussion of how to represent our formulas by *Gödel numbers*.

2.1 Formation rules for formulas

We are now in a position to make precise our formal equivalent of a mathematical statement, which we shall call a *formula* of our formal language. This usage differs from the everyday mathematical use of the word, where 'formula' usually means an expression such as $\sqrt{x^2+y^2}$ which takes a numerical value when numerical values are assigned to its variables. For us a formula is a formal counterpart of a mathematical statement. Here are two examples taken from Subsection 1.1.

See Example 1.1.

(a) $\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1)$

(b) $\forall x_0\, (\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1) \vee \exists x_1\, x_0 = ((\mathbf{0''} \cdot x_1) + \mathbf{0'}))$

Interpreting these formulas as referring to the natural numbers, (a) expresses 'x_0 is even' and (b) expresses 'every number is either even or odd'. With this interpretation, formula (b) is definitely true, whereas the truth of (a) depends on the value assigned to the variable x_0. For the moment, this difference is not important since our current concern is not with the meanings of these strings but with the formal rules which specify how they are formed.

Looking at the examples above we see that at their heart are equations, $x_0 = (\mathbf{0''} \cdot x_1)$ in (a) and (b), and also $x_0 = ((\mathbf{0''} \cdot x_1) + \mathbf{0'})$ in (b). The entire strings are built up from these equations between terms using the connectives and the quantifiers. We call these basic building blocks *atomic formulas*.

Definition 2.1 Atomic formula

A string is an *atomic formula* if and only if it has the form

$$\tau_1 = \tau_2$$

where τ_1 and τ_2 are terms.

In other words, the atomic formulas are obtained by taking two terms (which could be identical) and placing the identity symbol, =, between them.

Example 2.1

(a) The following are all atomic formulas:

(i) $x_0 = \mathbf{0}$

(ii) $x_3 = x_3$

(iii) $((x_1 \cdot x_2) + (x_1 \cdot x_3)) = (x_1 \cdot (x_2 + x_3))$

(iv) $x_2' = x_1$

This follows because each string contains just one occurrence of the symbol = and the strings to the left and the right of this symbol, namely x_0, $\mathbf{0}$, x_3, $((x_1 \cdot x_2) + (x_1 \cdot x_3))$, $(x_1 \cdot (x_2 + x_3))$, x_2', x_1, are all terms.

(b) The following are not atomic formulas:

(i) $\mathbf{0} = \mathbf{0} \cdot \mathbf{0}$

(ii) $\exists x_1\, x_1 = \mathbf{0}'$

(iii) $((x_1 \cdot x_1) + (x_2 \cdot x_2))$

(iv) $(x_1 = \mathbf{0}'' \vee x_1 = \mathbf{0}''')$

In cases (i) and (ii) the string contains just one occurrence of =, but in (i) the string to the right of it, namely $\mathbf{0} \cdot \mathbf{0}$, is not a term (as it lacks a pair of outer brackets), and in (ii) the string on the left, namely $\exists x_1\, x_1$, is not a term. String (iii) is not an atomic formula because it does not contain an occurrence of =, and string (iv) is not an atomic formula as it contains more than one occurrence of this symbol. ♦

It should be fairly clear that there is an algorithm for determining whether or not a given string is an atomic formula. First check that the given string has exactly one occurrence of the symbol =. If it doesn't, then the string is not an atomic formula. If it does, analyse the strings on either side of the symbol = to see if they are both terms. If they are both terms, then the original string is an atomic formula; if at least one is not a term, then the original string is not an atomic formula.

Problem 2.1

Which of the following are atomic formulas?

(a) $(x_1 + x_2)' = (x_1 + x_2')$

(b) $(x_6'' \cdot (x_{17} + \mathbf{0}''))' = (\mathbf{0} + \mathbf{0})$

(c) $(x_1 \cdot x_2) = (x_2 \cdot x_1) = x_0$

(d) $x_4 =$

(e) $x_0 \cdot \mathbf{0} = \mathbf{0} \cdot x_0$

Now we describe the set of all *formulas* of our formal language. The idea is that we allow as formulas exactly those strings of basic symbols which 'make mathematical sense' when we read $\forall$ as 'for all', $\exists$ as 'there exists', & as 'and', $\vee$ as 'or', $\neg$ as 'not', $\rightarrow$ as 'implies', $\leftrightarrow$ as 'if and only if' and = as 'equals'. However, as we have pointed out before, we must be precise enough so that there is an algorithm for determining whether or not a given string is a formula, so the above informal idea will not do as a definition. Instead, we give a definition, of the same character as our definition of terms, which is based on the idea that we build up formulas from atomic formulas using the connectives and quantifiers.

Definition 2.2 Formula

A string ϕ of basic symbols is a *formula* if and only if there is a finite sequence of strings $\phi_1, \phi_2, \ldots \phi_m$ of basic symbols such that, for $i < m$, each ϕ_i is a substring of ϕ, the string ϕ_m is ϕ and for $1 \leq i \leq m$ one of the following holds:

(a) ϕ_i is an atomic formula;

(b) for some $j < i$, ϕ_i is $\neg\phi_j$;

(c) for some $j, k < i$, ϕ_i is $(\phi_j \,\&\, \phi_k)$ or $(\phi_j \vee \phi_k)$ or $(\phi_j \rightarrow \phi_k)$ or $(\phi_j \leftrightarrow \phi_k)$;

(d) for some $j < i$ and some variable v, ϕ_i is $\forall v\, \phi_j$ or $\exists v\, \phi_j$.

ϕ is the Greek letter 'phi'.

As with the definition of terms, we have added the restriction that every string in the sequence is a substring of the final string ϕ in order to avoid at least some redundancy.

Just as for the symbols $+, \cdot$ appearing in a term, each appearance of one of $\&, \vee, \rightarrow, \leftrightarrow$ in a formula requires a corresponding pair of outer brackets.

For example, the sequence

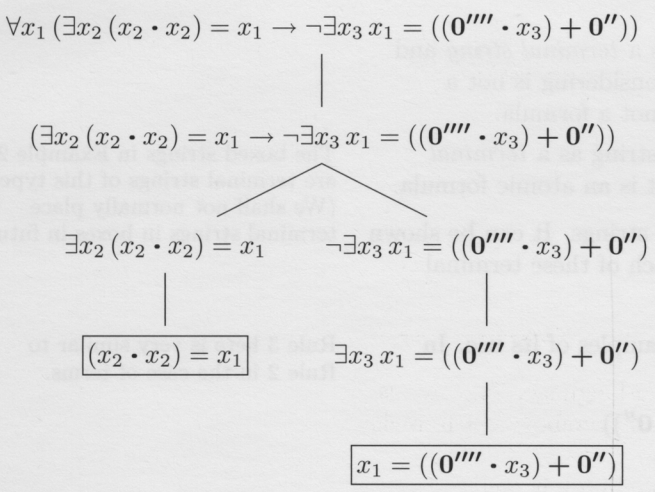
$$x_0 = (\mathbf{0}'' \cdot x_1), \quad x_0 = ((\mathbf{0}'' \cdot x_1) + \mathbf{0}'), \quad \exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1),$$
$$\exists x_1\, x_0 = ((\mathbf{0}'' \cdot x_1) + \mathbf{0}'),$$
$$(\exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1) \vee \exists x_1\, x_0 = ((\mathbf{0}'' \cdot x_1) + \mathbf{0}')),$$
$$\forall x_0\, (\exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1) \vee \exists x_1\, x_0 = ((\mathbf{0}'' \cdot x_1) + \mathbf{0}'))$$

satisfies the conditions specified by Definition 2.2. It follows that the final string in the list is a formula.

Each of these strings is also a formula. For a general formula ϕ, it can be shown that each of the strings ϕ_i in a list showing that ϕ is a formula is also a formula.

Because our definition of formulas has the same structure as the earlier definition of terms, we can analyse strings of basic symbols to determine whether they are formulas in exactly the same way as we did for terms. Before specifying explicitly how we carry out this analysis, we give an example.

Example 2.2



Note that the strings in boxes do not contain any connectives or quantifiers. Each of these two boxed strings contains just one occurrence of the identity symbol $=$, and it can be checked using the methods of the previous section that each has the form $\tau_1 = \tau_2$ where τ_1, τ_2 are terms. So both of the boxed strings are atomic formulas. The analysis shows how the string at the top is built up from these atomic formulas using connectives and quantifiers. Hence the analysis shows that the initial string is a formula. ♦

We now list explicitly the rules that we use in carrying out this analysis. As with terms, we should expect brackets to play a vital role. The step that we carry out at each point of the analysis depends on the first symbol in the string we are currently considering.

The rules are very similar to those for terms.

1 If the first symbol is the negation symbol then we write, under the given string, the string which results from it by deleting this negation symbol. That is, under a string of the form $\neg\phi$ we write the string ϕ.

2 If the first symbol is a quantifier followed by a variable then we write, under the given string, the string which results from it by deleting the quantifier and the variable. That is, under a string of the form $\forall v\,\phi$ or $\exists v\,\phi$, where v is a variable, we write the string ϕ.

3 If the first symbol is a left-hand bracket, the last symbol is a right-hand bracket and the string includes at least one of the connectives $\& \vee \rightarrow \leftrightarrow$, then we use the same bracket-counting method as we used with terms. The only difference is that now the *a-symbols* are the brackets and the connectives $\& \vee \rightarrow \leftrightarrow$. If this bracket count shows that there are equal numbers of left-hand and right-hand brackets, and there is just one bracket which is associated with the number 1 and is such that the next a-symbol following this bracket is one of the connectives $\& \vee \rightarrow \leftrightarrow$, then this connective is called the *principal connective* of the string. In this case the string has the form $(\phi * \psi)$, where $*$ is the principal connective, and we write the pair of strings ϕ, ψ below it.

ψ is the Greek letter 'psi'.

The bracket count may reveal in one of two ways that the string we are considering is not a formula.

(a) It may show that there are unequal numbers of left-hand and right-hand brackets.

(b) It may show that there is no principal connective, either because there is no bracket associated with the number 1 such that the next a-symbol is one of the connectives $\& \vee \rightarrow \leftrightarrow$, or because there is more than one such bracket.

In either of these cases we describe the string as a *terminal string* and our analysis can terminate. The string we are considering is not a formula, and it follows that the initial string is not a formula.

4 If none of Rules 1 to 3 applies, we describe the string as a *terminal string* and we analyse it to determine whether it is an atomic formula.

The boxed strings in Example 2.2 are terminal strings of this type. (We shall not normally place terminal strings in boxes in future.)

All branches of the analysis must end with terminal strings. It can be shown that the original string is a formula if and only if each of these terminal strings is an atomic formula.

Since Rule 3 is rather complicated, we give some examples of its use. In Example 2.2 we applied it to the string

Rule 3 here is very similar to Rule 2 in the case of terms.

$$(\exists x_2\,(x_2 \cdot x_2) = x_1 \rightarrow \neg\exists x_3\,x_1 = ((\mathbf{0''''} \cdot x_3) + \mathbf{0''}))$$

The bracket count for this string is as follows:

$$\begin{array}{cccccccccccccccc} (& \exists x_2 & (& x_2 \cdot x_2 &) = & x_1 & \rightarrow & \neg\exists x_3 & x_1 & = & (& (& \mathbf{0''''} \cdot x_3 &) + & \mathbf{0''} &) \\ 1 & & 2 & & 1 & & & & & & 23 & & & 2 & & 10 & \end{array}$$

This shows that there are equal numbers of left-hand and right-hand brackets, and that $\rightarrow$ is the principal connective. Thus this string has the form $(\phi \rightarrow \psi)$ where ϕ is the string $\exists x_2\,(x_2 \cdot x_2) = x_1$ and ψ is the string $\neg\exists x_3\,x_1 = ((\mathbf{0''''} \cdot x_3) + \mathbf{0''})$, and hence our analysis proceeded by analysing the two strings ϕ and ψ.

Example 2.3

(a) $\underset{1}{(}\exists x_1\ \underset{2}{(}x_1 = \underset{3}{(}x_2 \cdot x_2\underset{2}{)} \rightarrow \neg\exists x_3\, x_1 = x_3'\underset{10}{)}\underset{}{)}$

The bracket count shows that there is no bracket associated with the number 1 such that the next a-symbol is a connective. Hence this string is not a formula.

(b) $\underset{1}{(}\exists x_0\, \exists x_1\, x_2 = \underset{2}{(}x_0 + x_1\underset{1}{)'} \rightarrow \exists x_0\, x_2 = \underset{2}{(}x_0 + \mathbf{0''}\underset{1}{)} \vee x_2 = \mathbf{0}\underset{0}{)}$

There are two brackets associated with the number 1 such that the next a-symbol is a connective. Hence this string does not have a principal connective and so is not a formula. ♦

Problem 2.2

In each of the following cases determine whether the given string has a principal connective.

(a) $(x_0 = \mathbf{0'} \vee (x_0 = (\mathbf{0'} \cdot x_1) \rightarrow x_0 = (x_1 \cdot x_1)))$

(b) $(\exists x_1\, ((x_1 \cdot x_1) = x_0)\ \&\ x_0 = ((x_1 \cdot \mathbf{0''}) + \mathbf{0'}))$

(c) $(((x_0 + x_0) + x_1') = (x_2 + x_0) \vee (x_0 \cdot x_0) = x_1 \rightarrow \neg\exists x_2\, (x_2 \cdot x_1) = x_0)$

(d) $(\exists x_0\, ((x_1 = (x_0 \cdot x_2)) \rightarrow x_1 = (x_2 \cdot x_2)))$

Problem 2.3

Analyse each of the following strings and hence decide which of them are formulas.

(a) $\forall x_1\, (\exists x_2\, (x_1 \cdot x_1) = (x_2 + x_2) \rightarrow \exists x_3\, x_1 = (x_3 + x_3))$

(b) $\forall x_0\, \neg\exists x_1\, (x_1 \cdot x_1) = x_0$

(c) $((\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \vee x_1 = \mathbf{0}) \vee x_1 = \mathbf{0'})$

(d) $(x_0 = \mathbf{0} \vee x_1 = \mathbf{0}\ \&\ x_2 = \mathbf{0})$

(e) $(\exists x_1\, ((x_1 \cdot x_1) = x_0\ \&\ x_0 = ((x_1 \cdot \mathbf{0''}) + \mathbf{0'}))$

Problem 2.4

In Subsection 1.1 we obtained the following expression for Euclid's Theorem in terms of basic symbols:

$$\forall x_0\, \exists x_1\, ((\neg\, x_1 = \mathbf{0'}\ \&\ \forall x_2\, (\exists x_3\, (x_2 \cdot x_3 = x_1) \rightarrow (x_2 = \mathbf{0'} \vee x_2 = x_1)))\ \&\ \exists x_4\, (x_0 + x_4 = x_1))$$

Adjust this expression so that it is a formula of our formal language.

For technical reasons that will become apparent later, we now introduce the idea of a subformula.

> ***Definition 2.3 Subformula***
>
> The formulas that occur in the analysis of a given formula ϕ are called the *subformulas* of ϕ.

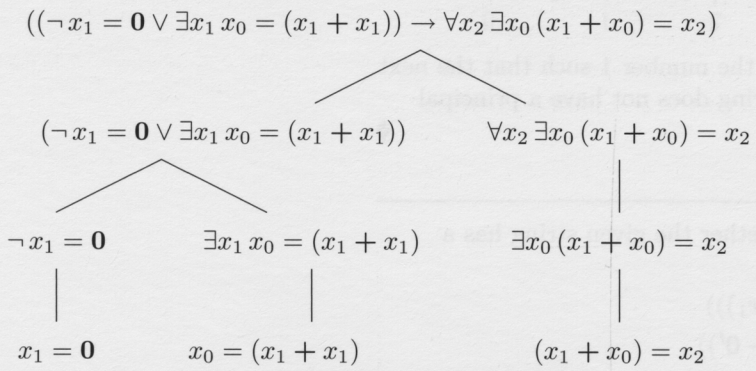
Notice that the analysis of a formula ϕ begins with ϕ itself, and so ϕ is counted as one of its own subformulas. This conforms with the usual mathematical usage of the prefix 'sub', as, for example, in the terminology 'subset' and 'subgroup'.

Example 2.4

Let us find the subformulas of the formula

$$((\neg x_1 = \mathbf{0} \vee \exists x_1\, x_0 = (x_1 + x_1)) \rightarrow \forall x_2 \exists x_0\, (x_1 + x_0) = x_2)$$

We begin with the analysis of this string which shows that it is a formula. The bracket count shows that $\rightarrow$ is the principal connective, and the full analysis is as follows:



We see from this analysis that the string is indeed a formula as all the terminal strings are atomic formulas. The subformulas of the initial formula are the formulas which occur in this analysis, that is, they are as follows:

$x_1 = \mathbf{0}$

$x_0 = (x_1 + x_1)$

$(x_1 + x_0) = x_2$

$\neg x_1 = \mathbf{0}$

$\exists x_1\, x_0 = (x_1 + x_1)$

$\exists x_0\, (x_1 + x_0) = x_2$

$(\neg x_1 = \mathbf{0} \vee \exists x_1\, x_0 = (x_1 + x_1))$

$\forall x_2 \exists x_0\, (x_1 + x_0) = x_2$

$((\neg x_1 = \mathbf{0} \vee \exists x_1\, x_0 = (x_1 + x_1)) \rightarrow \forall x_2 \exists x_0\, (x_1 + x_0) = x_2)$ ♦

Problem 2.5

List the subformulas of the following formulas.

(a) $\forall x_0 \neg \exists x_1\, (x_1 \cdot x_1) = x_0$

(b) $((\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \vee x_1 = \mathbf{0}) \vee x_1 = \mathbf{0'})$

(c) $x_1 = ((x_2 \cdot x_3) + \mathbf{0})$

(a) is the formula of Problem 2.3(b) and (b) is the formula of Problem 2.3(c).

2.2 Gödel numbers

Leibniz's Question, you will recall, asks whether there is an algorithm for deciding which statements of number theory are true. We are now in a position to say that by a 'statement' we mean a statement that can be expressed by a formula of our formal language. And we have seen that there are algorithms for determining whether or not a string of symbols is a term or a formula, which is a first step towards our goal of determining whether there is an algorithm for deciding the truth of statements.

We shall complete our investigation of whether there is such an algorithm in *Unit 8.*

In *Units 1* to *3*, we related algorithms to URM-computable functions, which map natural numbers, or ordered k-tuples of natural numbers, to natural numbers. To make another step towards our goal, we need to see how our algorithms which apply to strings of symbols can be related to URM-computable functions. We do this by showing how strings of symbols can be coded by natural numbers, and hence how formulas can be coded in this way.

Our method for assigning numbers, which are called *Gödel numbers*, to formulas is based on the same principle as our method for coding URM programs. A formula is a finite string of basic symbols, just as a URM program is a finite sequence of URM instructions. So we assign Gödel numbers to formulas by first assigning numbers to the basic symbols and then coding finite strings of these symbols. For definiteness we have to describe a particular way to do this. It should be evident, however, that the particular details are not that important. The key properties that we need our coding to satisfy are:

This use of coding originated in Gödel's work and is named after him. The coding of URM programs was discussed in *Unit 2*.

(a) different strings of symbols get assigned different Gödel numbers;

(b) given a natural number, there is an algorithm for deciding whether it codes a string of basic symbols and which string it codes;

(c) given a string of symbols, there is an effective method for working out its Gödel number.

We now describe the particular Gödel numbering that we have chosen to use. We begin by specifying which numbers are assigned to the basic symbols:

Symbol	(	)	&	$\vee$	$\rightarrow$	$\leftrightarrow$	$\neg$	$\forall$	$\exists$	**0**	$'$	$+$	$\cdot$	$=$	x_i
Gödel number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	$i+15$

Note that each positive integer codes a basic symbol, and that, in particular, integers ≥ 15 code variables. We use the notation $\beta(\sigma)$ to indicate the number which codes the symbol σ. For example $\beta(\forall) = 8$ and $\beta(x_{27}) = 42$.

Corresponding to each string $\sigma_1\sigma_2\ldots\sigma_k$ of basic symbols we have the sequence $\beta(\sigma_1), \beta(\sigma_2), \ldots, \beta(\sigma_k)$ of Gödel numbers. We then assign to the string the Gödel number

$$p_1^{\beta(\sigma_1)} p_2^{\beta(\sigma_2)} \ldots p_k^{\beta(\sigma_k)}$$

where as usual $p_1, p_2, \ldots, p_k$ are the first k prime numbers. If ϕ is a string of basic symbols, we use the notation $\gamma(\phi)$ to indicate its Gödel number.

Notice that we use exactly the same prime power method for mapping the sequence of numbers $\beta(\sigma_1), \beta(\sigma_2), \ldots, \beta(\sigma_k)$ to a single number as we did in *Unit 2* when we were coding URM programs.

For example, if ϕ is the string $\forall x_0\, x_0 = x_0$ then we obtain the sequence $8, 15, 15, 14, 15$ of Gödel numbers for the basic symbols which make up the string. Hence the Gödel number of this string is

$$2^8 3^{15} 5^{15} 7^{14} 11^{15}.$$

This is a very large number which contains 48 digits when written out in decimal notation. Written in that way it would be hard for us to see immediately that it was the Gödel number of the string ϕ. But, in principle, it is straightforward, given a positive integer, to work out its prime decomposition and hence which string, if any, it represents.

The string $\forall x_0\, x_0 = x_0$ whose Gödel number we have calculated is a formula. But Gödel numbers are defined for all strings of basic symbols, whether or not they are formulas. For example, the string $\forall = x_1 \neg x_3$ which is definitely not a formula has the Gödel number

$$2^8 3^{14} 5^{16} 7^7 11^{18}.$$

Problem 2.6

Find the Gödel numbers (expressed in terms of their prime decompositions) of the following formulas.

(a) $x_3 = ((x_2 \cdot \mathbf{0''}) + x_0)$

(b) $\forall x_0\,(x_0' = \mathbf{0'} \leftrightarrow x_0 = \mathbf{0})$

Problem 2.7

Determine whether each of the following is the Gödel number of a string of basic symbols and, if so, specify the relevant string.

(a) $2^7 3^{19} 5^{14} 7^{16} 11^2$

(b) $2^8 3^{15} 5^9 7^{16} 11^{15} 13^{11} 17^{14} 23^{16}$

(c) $2^8 3^{15} 5^9 7^{16} 11^{15} 13^{11} 17^{14} 19^8$

The existence of an algorithm for determining which strings are formulas can now be restated in terms of Gödel numbers by saying that there is an algorithm for computing whether a natural number is the Gödel number of a formula. Using Church's Thesis, this means that the set of Gödel numbers of formulas is recursive. (In fact this set is primitive recursive, but we shall not show this in the course.)

Recall that by this we mean that the characteristic function of the set is recursive.

The representation of the formulas of our formal language by Gödel numbers will be exploited in a very subtle way. Formulas of our language are naturally interpreted as expressing properties of numbers; but, when formulas are coded by numbers, we can also interpret formulas as expressing properties of formulas. This sort of self-reference lies at the heart of the proof of Gödel's Theorems and we explain it in detail in *Unit 8*.

Notational convention

We have specified that a variable is one of $x_0, x_1, x_2 \ldots$. An infinite list of variables is needed because, although each formula is a finite string of symbols and can only include a finite number of different variables, we cannot impose any upper bound on what this number might be. The use of the numerical subscripts makes it easy for us to assign Gödel numbers to the variables. The disadvantage of this notation is that it makes formulas less easy to read than if different non-subscripted letters such as $x, y, z, t, \ldots$ were to be used. Henceforth, to make formulas more easy to read, we shall fall in line with this common mathematical usage and write $x, y, z, t, \ldots$ for variables. If you like, you can regard these symbols as abbreviations for $x_0, x_1, x_2, x_3, \ldots$. All that really matters is that they represent distinct variables.

3 INTERPRETATIONS

Leibniz's Question is about deciding which statements of number theory are true, and so it involves the meanings of mathematical assertions. But in specifying the formation rules of our language we have at no stage needed to use the intended meaning of our symbols. We have used familiar symbols such as $=$, $+$, & to indicate the meaning we have in mind, but the intended meanings play no part in our definitions of terms and formulas. For example, we decide whether a given string includes a principal connective by our bracket-counting method, in which the intended meaning of the connective is irrelevant. At some point, however, we need to make a connection between the strings of basic symbols which count as formulas and the statements of number theory which they can be interpreted as expressing. We have reached that point now.

As we shall see in this section, the logical symbols have fixed interpretations, but the interpretation of the arithmetical symbols will vary according to the system of numbers or other objects to which we are applying our formal language. In addition we shall use symbol interpretation to make more precise what it means for one statement to be inferred from others, using the notion of *logical consequence*. This will later help us towards resolving Hilbert's Question, which requires a description of what we mean by a proof.

3.1 The interpretation of the logical connectives

In this subsection we shall deal with the interpretation of the logical connectives: $\neg$, $\&$, $\vee$, $\rightarrow$ and $\leftrightarrow$. First we need to understand what is 'logical' about them. This is best done by looking at examples from ordinary English.

Consider the following three arguments.

(1) If my horse races on soft ground then he will win.
My horse is racing today and the ground is soft.
Therefore my horse will win today.

(2) She hit the ball with spin and she hit it hard.
Therefore she hit the ball hard.

(3) All men are mortal.
Socrates is a man.
Therefore Socrates is mortal.

All three are valid arguments, but from where does this validity derive? Not from the actual subject-matter of the statements. No knowledge of the turf, tennis or Socrates is required to see that the above statements form valid arguments. Rather their validity derives solely from the normal meanings of the English 'connecting' words 'if ... then ...', 'and', 'all' and so on. Thus the truth of the statement 'She hit the ball with spin and she hit it hard' *guarantees* the truth of the statement 'she hit the ball hard' simply because of the meaning of the word 'and' and not because of the meanings of 'hit', 'ball', 'spin' or 'hard'. Indeed, if A, B represent two statements, then,

if the statement

A and B

is true, *then* the statement

B

must also be true (as also must A), no matter what the meanings of the statements A and B.

This shows the context-free nature of argument (2). Similarly (1) and (3) are just particular cases of generally valid arguments. Argument (1) just uses the meaning of 'if ... then ...' from which it follows that,

if the statements

if A then B

and

A

are true, *then* the statement

B

must also be true, no matter what the meanings of the statements A and B.

Similarly, argument (3) has the form

if all Xs are Y *and* Z is an X
then Z is a Y

which is valid not because of the subject-matter, since there is none in this abstract version, but because of the meanings of 'all', 'are' and 'is'.

Logic is the study of valid forms of reasoning: that is, forms of argument which are valid because of their structure, irrespective of the content. As we have seen this validity arises from the meanings of certain key logical words such as 'and' and 'if ... then ...'. Included among the logical symbols of our formal language are symbols which correspond to these key words. We now set about describing precisely how these logical connectives are to be interpreted. We start with &, followed by $\neg$, $\vee$, $\rightarrow$ and $\leftrightarrow$ in turn.

In what follows, we shall use the term *given interpretation* to mean the system of numbers (or other objects) to which our language is being applied together with a corresponding interpretation of the arithmetic symbols.

There is a more formal definition of the term *interpretation* in Subsection 3.2.

& (conjunction)

If ϕ and ψ are formulas, then the string $(\phi \,\&\, \psi)$ is also a formula, and we call it the *conjunction* of ϕ and ψ. The interpretation of the symbol & is explained by specifying how we work out whether or not the conjunction $(\phi \,\&\, \psi)$ is true under a given interpretation in terms of what we know about the truth of ϕ and the truth of ψ under the same interpretation.

Also we call a formula of the form $(\phi \,\&\, \psi)$ a *conjunction.*

> The formula $(\phi \,\&\, \psi)$ is true in a given interpretation if and only if both the formulas ϕ and ψ are true in this interpretation.

Thus we interpret the symbol & so that it corresponds to the use of 'and' in everyday English.

Another way to explain the interpretation of & is by means of the following table.

ϕ	ψ	$(\phi \,\&\, \psi)$
true	true	true
true	false	false
false	true	false
false	false	false

This table specifies the truth of the formula $(\phi \,\&\, \psi)$ for all possible combinations of the truth and falsity of the formulas ϕ and ψ. We call it the *truth table* for the symbol &. We describe each of 'true' and 'false' as a *truth value.*

The following example should make things even clearer. Suppose that ϕ is the formula $\exists x\, x' = \mathbf{0}$ and that ψ is the formula $\exists x\, (x \cdot x) = \mathbf{0}''$. The truth of these formulas depends on how we interpret the arithmetical symbols $\mathbf{0}$ $'$ $\cdot$ and which system of numbers (or other objects) we have in mind. For example, if we interpret $\mathbf{0}$ as referring to the number 0, $'$ as meaning 'add one' and $\cdot$ as multiplication, then ϕ is a true statement about the set $\mathbb{Z}$ of integers (as there is an integer $x \in \mathbb{Z}$ with $x + 1 = 0$, namely $x = -1$) but it is a false statement about the set $\mathbb{N}$ of natural numbers (as -1 is not in $\mathbb{N}$). Similarly, keeping the same interpretations of the arithmetical symbols, ψ is true when we interpret it as being about a number system which includes a square root of 2.

However, and this is the key point, once we know whether or not the formulas ϕ and ψ are true in a given interpretation, we can immediately work out from the truth table whether or not the conjunction $(\phi \,\&\, \psi)$ is true. At this stage the nature of the given interpretation is irrelevant: all that matters is whether ϕ and ψ turn out to be true.

Exactly similar remarks will apply to the truth tables for the other logical connectives. So for the purpose of this discussion we can ignore the given interpretation.

For technical reasons it is more convenient to use 1 and 0 as truth values. 1 replaces 'true' and 0 replaces 'false', so that the truth table for conjunction would be written as follows.

ϕ	ψ	$(\phi \,\&\, \psi)$
1	1	1
1	0	0
0	1	0
0	0	0

We make one further simplification which will be useful later on. We often abbreviate the above table in the following way.

$(\phi$	$\&$	$\psi)$
1	1	1
1	0	0
0	0	1
0	0	0

The truth values of the constituent formulas ϕ, ψ are written in columns under the formulas, and the corresponding truth value of the conjunction $(\phi \,\&\, \psi)$ is written in a column beneath the symbol &.

¬ (negation)

If ϕ is a formula then the string $\neg\phi$ is also a formula, called the *negation* of ϕ. The interpretation of the symbol ¬ is given by the following truth table.

Also we call a formula of the form $\neg\phi$ a *negation*.

ϕ	$\neg\phi$
1	0
0	1

or in abbreviated form

$\neg$	ϕ
0	1
1	0

Thus we interpret the symbol ¬ so that it corresponds to the use of 'not' in everyday English. However, whereas in standard English we form the negation of a sentence by inserting the word 'not' in the appropriate place within the sentence, in our formal language we negate a formula by prefixing the symbol ¬ to the string which makes up the formula. For example the negation of

The use of the symbol ¬ derives from the notation Frege used for negation; however, our symbolism is, in other respects, very different from his.

'x is an even number' is 'x is not an even number'

but the negation of the formula

$\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1)$ is $\neg\exists x_1\, x_0 = (\mathbf{0''} \cdot x_1)$

$\vee$ (disjunction)

If ϕ, ψ are formulas, then the string $(\phi \vee \psi)$ is also a formula which is called the *disjunction* of ϕ and ψ. The truth table for $\vee$ is as follows.

Also we call a formula of the form $(\phi \vee \psi)$ a *disjunction.*

ϕ	ψ	$(\phi \vee \psi)$
1	1	1
1	0	1
0	1	1
0	0	0

or in abbreviated form

$(\phi$	$\vee$	$\psi)$
1	1	1
1	1	0
0	1	1
0	0	0

Thus the interpretation of the symbol $\vee$ corresponds to the everyday use of 'or' in its inclusive sense (in which 'A or B' means 'A or B or both').

The use of the symbol $\vee$ derives from the Latin word 'vel', as the construction 'vel A vel B' is used in Latin for the inclusive 'A or B'.

$\rightarrow$ (implication)

If ϕ, ψ are formulas, then so also is the string $(\phi \rightarrow \psi)$ and we call it the *implication* with *antecedent* ϕ and *consequent* ψ. The truth table for $\rightarrow$ is as follows.

Also we call a formula of the form $(\phi \rightarrow \psi)$ an *implication.*

ϕ	ψ	$(\phi \rightarrow \psi)$
1	1	1
1	0	0
0	1	1
0	0	1

or in abbreviated form

$(\phi$	$\rightarrow$	$\psi)$
1	1	1
1	0	0
0	1	1
0	1	0

Note that the only case where $(\phi \rightarrow \psi)$ is false is where the antecedent ϕ is true and the consequent ψ is false. This might seem rather strange, and so we add some remarks to justify the claim that the above truth table does reflect the usage of 'if ... then ...' in mathematics (which is rather different from its use outside mathematics).

Consider the following fairly typical use of an 'if ... then ...' statement in mathematics.

$$\text{For all } x \in \mathbb{R}, \text{ if } x > 3 \text{ then } x^2 > 9. \tag{3.1}$$

We cannot say whether the antecedent and consequent of the implication

$$\text{if } x > 3 \text{ then } x^2 > 9 \tag{3.2}$$

are true until we know what value is assigned to the variable x. The point of assertion (3.1) is to say that this implication is true *whatever* real number value is assigned to x. Our truth table for the connective $\rightarrow$ must be consistent with this, even though when we give x a specific value we may obtain an implication which, although true, may seem a little unnatural. Thus suppose, for example, that we successively give x the values 7, 2, -5 in (3.2). Since (3.1) is always true, the implications obtained from (3.2) in this way must be true. These implications are:

(a) If $7 > 3$ then $49 > 9$.

(b) If $2 > 3$ then $4 > 9$.

(c) If $-5 > 3$ then $25 > 9$.

In (a) both the antecedent and the consequent are true. Thus (a) corresponds to the first row of our truth table. In both (b) and (c) the antecedent is false. In (b) the consequent is also false and so (b) corresponds to the bottom row of the truth table. In (c) the consequent is true, so this implication corresponds to row 3 of the truth table. These last two examples help to explain why rows 3 and 4 of the truth table for $\rightarrow$ take the given form.

$\leftrightarrow$ (bi-implication)

If ϕ, ψ are formulas, then the string $(\phi \leftrightarrow \psi)$ is also a formula called the *bi-implication* of ϕ and ψ. The truth table for $\leftrightarrow$ is as follows.

Also we call a formula of the form $(\phi \leftrightarrow \psi)$ a *bi-implication.*

ϕ	ψ	$(\phi \leftrightarrow \psi)$
1	1	1
1	0	0
0	1	0
0	0	1

or in abbreviated form

$(\phi$	$\leftrightarrow$	$\psi)$
1	1	1
1	0	0
0	0	1
0	1	0

The symbol $\leftrightarrow$ corresponds to the use of the expression 'if and only if' in mathematics. Notice that $(\phi \leftrightarrow \psi)$ takes the value 1 exactly when ϕ and ψ have the same truth values.

An alternative way of thinking about 'ϕ if and only ψ' is as 'ϕ implies ψ and ψ implies ϕ'. You can check that $(\phi \leftrightarrow \psi)$ and $((\phi \rightarrow \psi)\ \&\ (\psi \rightarrow \phi))$ have the same truth table in Additional Exercise 2 for this section.

Now that we have the basic truth tables for the logical connectives, we can use them to work out truth tables for more complicated formulas built up using the connectives. We describe the general method and then give some examples.

Suppose that we have a formula θ built up from some of its subformulas $\phi_1, \phi_2, \ldots, \phi_k$ using only connectives. To get one row of the truth table for θ, we assign a truth value to each of $\phi_1, \phi_2, \ldots, \phi_k$ and progressively work out the truth values of all the longer subformulas of θ corresponding to this assignment of truth values using the rules given in the five truth tables for the logical connectives. The order in which we deal with the subformulas corresponds to the way in which θ is built up as a formula from $\phi_1, \phi_2, \ldots \phi_k$. Finally we arrive at a truth value for θ (which is by definition a subformula of itself). Then we repeat this process for all possible assignments of truth values to $\phi_1, \phi_2, \ldots, \phi_k$ to obtain the full truth table for θ.

θ is the Greek letter 'theta'.

Example 3.1

Consider the formula θ given by

$$(\neg(x' = \mathbf{0} \vee \exists x\,(x \cdot x) = \mathbf{0}'') \rightarrow (x' = \mathbf{0}\ \&\ \exists x\,(x \cdot x) = \mathbf{0}''))$$

If ϕ is the subformula $x' = \mathbf{0}$ and ψ is the subformula $\exists x\,(x \cdot x) = \mathbf{0}''$ then θ is the formula

$$(\neg(\phi \vee \psi) \rightarrow (\phi\ \&\ \psi))$$

We can work out its truth table as follows.

ϕ	ψ	$(\phi \vee \psi)$	$\neg(\phi \vee \psi)$	$(\phi\ \&\ \psi)$	$(\neg(\phi \vee \psi) \rightarrow (\phi\ \&\ \psi))$
1	1	1	0	1	1
1	0	1	0	0	1
0	1	1	0	0	1
0	0	0	1	0	0

The headings of the columns correspond to the subformulas we obtain when building up the formula $(\neg(\phi \vee \psi) \rightarrow (\phi\ \&\ \psi))$ from ϕ and ψ. The first two columns list all possible combinations of truth values that can be attained by ϕ and ψ. In the third column the corresponding truth values for $(\phi \vee \psi)$ are computed according to the truth table for disjunction. In the fourth column we have applied the truth table for negation to the truth values in the third column. This gives us the corresponding truth values for the formula $\neg(\phi \vee \psi)$. The fifth column is computed using the truth table for conjunction applied to the first two columns. Finally, we work out the sixth column by applying the truth table for implication to columns four and five. This gives us the truth values of the formula $(\neg(\phi \vee \psi) \rightarrow (\phi\ \&\ \psi))$. For example, the truth table tells us that when ϕ is true and ψ is false (in the second row of the truth table), then the entire formula is true. ♦

Note that, on each row of the truth table in Example 3.1, the truth value of the formula θ is completely determined by the truth values given to the subformulas ϕ and ψ, and the rules given by the five basic truth tables. This is guaranteed by the strict way in which bracketing was stipulated when we gave the definition of a formula. If we had been sloppy in this definition we might have allowed a string such as $(\phi \,\&\, \psi \vee \chi)$ to count as a formula, but this does not have a well-defined truth table, since it could be calculated as though the 'formula' were $((\phi \,\&\, \psi) \vee \chi)$ or as though it were $(\phi \,\&\, (\psi \vee \chi))$, and these two formulas have different truth tables.

You might like to check that $((\phi \,\&\, \psi) \vee \chi)$ and $(\phi \,\&\, (\psi \vee \chi))$ have different truth tables in Additional Exercise 3 for this section.

The truth table of Example 3.1 can be written in a more succinct form as follows.

$(\neg$	$(\phi$	$\vee$	$\psi)$	$\rightarrow$	$(\phi$	$\&$	$\psi))$
0	1	1	1	1	1	1	1
0	1	1	0	1	1	0	0
0	0	1	1	1	0	0	1
1	0	0	0	0	0	0	0
③	①	②	①	④	①	②	①

The circled numbers indicate the order in which the columns were filled in. They are not part of the truth table and they could be left out. The values in column ④ tell us the truth value of the entire formula corresponding to the truth values of the subformulas ϕ, ψ in the columns labelled ①. Note that column ④ is that in which the principal connective, here $\rightarrow$, of the entire formula occurs.

We shall include these circled numbers in all the truth tables in this subsection, including our solutions to problems. They are not expected within solutions to assessment questions, although they can be of practical assistance when constructing truth tables for complicated formulas.

The truth table has 4 rows, corresponding to the $4 = 2 \times 2$ possible combinations of truth values for the two subformulas ϕ and ψ. For a formula built up from three different subformulas using just the connectives, as in Example 3.2 below, there are $2 \times 2 \times 2 = 8$ possible combinations. Hence the truth table for such a formula has 8 rows. In general, the truth table for a formula built up from n different subformulas has 2^n rows.

The order in which we write the rows of a truth table does not matter. What matters is that all possible combinations of truth values for the subformulas are covered. However, to ensure that all cases are covered, it is best to list the combinations of truth values in a systematic way, as described in Example 3.2.

Example 3.2

We calculate the truth values of the formula

$$((\neg\phi \,\&\, \psi) \leftrightarrow \chi)$$

for all possible combinations of truth values for its subformulas ϕ, ψ, χ. Since there are three subformulas, the truth table has 8 rows.

ϕ	ψ	χ	$\neg\phi$	$(\neg\phi \,\&\, \psi)$	$((\neg\phi \,\&\, \psi) \leftrightarrow \chi)$
1	1	1	0	0	0
1	1	0	0	0	1
1	0	1	0	0	0
1	0	0	0	0	1
0	1	1	1	1	1
0	1	0	1	1	0
0	0	1	1	0	0
0	0	0	1	0	1

The abbreviated form of this truth table is as follows:

$((\neg$	ϕ	$\&$	$\psi)$	$\leftrightarrow$	$\chi)$
0	1	0	1	0	1
0	1	0	1	1	0
0	1	0	0	0	1
0	1	0	0	1	0
1	0	1	1	1	1
1	0	1	1	0	0
1	0	0	0	0	1
1	0	0	0	1	0
②	①	③	①	④	①

Our systematic method for listing all the different combinations of truth values is seen more clearly in the full table. It can be described in two different ways. In the column for ϕ we have entered the values 1, 0 in blocks of 4, in the ψ column they are in blocks of 2, and in the χ column there is alternately a 1 followed by a 0. Another way to look at it is to view the three truth values as representing a number written in binary notation, so that in the successive rows of the table the numbers corresponding to the truth values for ϕ, ψ, χ are $7, 6, 5, 4, 3, 2, 1, 0$. ♦

Problem 3.1

Work out the truth tables for the following formulas.

(a) $((\phi \,\&\, \psi) \,\&\, \chi)$

(b) $((\phi \rightarrow \psi) \rightarrow \phi)$

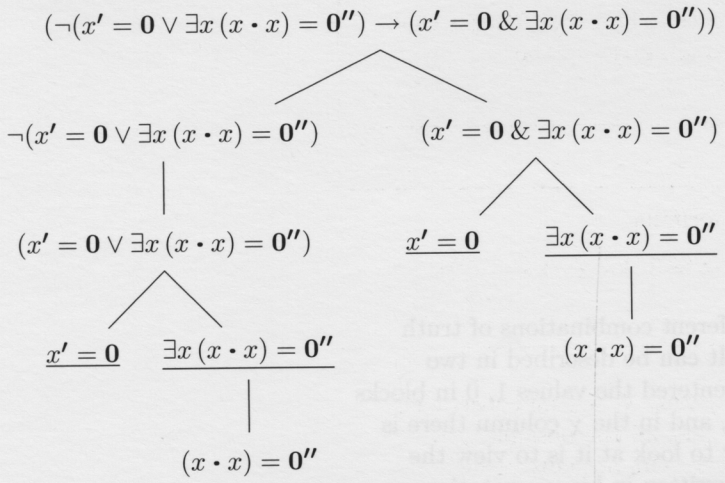
(c) $(\phi \leftrightarrow \phi)$

(d) $(\neg(\phi \vee \psi) \leftrightarrow \chi)$

We now turn to an important question: given a formula of our formal language, how do we decide which subformulas to use in its truth table analysis? Fortunately, the method which we described in Subsection 2.1 for analysing a string of basic symbols to determine whether it is a formula also answers this question. We carry out the process described in Subsection 2.1 except that, whenever we reach a string which begins with a quantifier, we do not analyse that string any further. If the original string is a formula, then the strings obtained in this way, which must either be atomic formulas or begin with a quantifier, will be the required subformulas. In this way, the subformulas used to construct a truth table are such that the original formula can be built up from them using only logical connectives. Furthermore, the subformulas themselves either begin with a quantifier or are atomic formulas.

Example 3.3

We consider the formula of Example 3.1. The full analysis which shows that it is a formula is as follows.



The terminal strings are all atomic formulas and so the original string is a formula. For the truth table analysis we stop analysing a branch when we find a string beginning with a quantifier but otherwise continue until we get to an atomic formula. We stop at the four underlined strings. Two of these begin with quantifiers: both are the subformula $\exists x\,(x \cdot x) = \mathbf{0}''$. The other two are atomic formulas: both are the subformula $x' = \mathbf{0}$. These are the subformulas which were used in the truth table of Example 3.1. ♦

Problem 3.2

Analyse each of the following formulas to find the subformulas to be used in their truth tables. Draw up the truth table for each formula.

(a) $(\forall x\,\exists y\,y' = x \rightarrow \exists y\,y' = \mathbf{0})$

(b) $(((\exists x\,(x + y) = \mathbf{0} \rightarrow y = \mathbf{0}') \,\&\, (\exists x\,(x + y) = \mathbf{0} \vee \forall z\,y = z)) \rightarrow (y = \mathbf{0}' \vee \forall z\,y = z))$

As can be seen from the solutions to Problems 3.1 and 3.2, there are some formulas which receive the truth value 1 whatever truth values are assigned to the subformulas from which they are built up. Here is another example.

$(((\phi$	$\rightarrow$	$\psi)$	$\&$	$(\psi$	$\rightarrow$	$\chi))$	$\rightarrow$	$(\neg$	χ	$\rightarrow$	$\neg$	$\phi))$
1	1	1	1	1	1	1	1	0	1	1	0	1
1	1	1	0	1	0	0	1	1	0	0	0	1
1	0	0	0	0	1	1	1	0	1	1	0	1
1	0	0	0	0	1	0	1	1	0	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0
0	1	1	0	1	0	0	1	1	0	1	1	0
0	1	0	1	0	1	1	1	0	1	1	1	0
0	1	0	1	0	1	0	1	1	0	1	1	0
①	②	①	③	①	②	①	④	②	①	③	②	①

The property of always receiving the truth value 1 is of great significance in our future work. Formulas with this property will be true however we interpret the arithmetical symbols which occur in them. They are called *tautologies*. The precise definition is as follows.

Definition 3.1 Tautology

Suppose that θ is a formula which is built up from subformulas $\phi_1, \phi_2, \ldots, \phi_k$ using only the logical connectives and that θ takes the truth value 1 no matter what truth values are given to the subformulas. Then θ is called a *tautology.*

Problem 3.3

Use a truth table analysis to show that the formula

$$(\forall x\, x = y \rightarrow (\neg \exists z\, (z + \mathbf{0''}) = y \rightarrow (\exists z\, (z + \mathbf{0''}) = y \rightarrow x = \mathbf{0'})))$$

is a tautology.

We end this subsection with rather a subtle point. When we construct a truth table for a formula, we tacitly assume that all the possible combinations of truth values for its subformulas can actually occur. However, since these subformulas have their own internal structure, this may not always be the case. We explain this point with reference to the formulas of Example 3.1 and Problem 3.2(a).

In the case of the formula of Example 3.1, the subformulas which we used in the truth table were $x' = \mathbf{0}$ and $\exists x\, (x \cdot x) = \mathbf{0''}$. Suppose we interpret the arithmetical symbols $\mathbf{0}$, $'$, $\cdot$ as referring to the number 0, the operation 'add one' and the operation of multiplication, respectively. Then $x' = \mathbf{0}$ is true whenever x is interpreted by the number -1. And the subformula $\exists x\, (x \cdot x) = \mathbf{0''}$ is true whenever it is interpreted as referring to a number system which includes $-\sqrt{2}$ or $+\sqrt{2}$. So either subformula can be true or false independently of the other subformula. Hence each row of the truth table corresponds to a combination of truth values that can occur.

The position is different for the formula of Problem 3.2(a). The subformulas which we use for its truth table are $\forall x\, \exists y\, y' = x$ and $\exists y\, y' = \mathbf{0}$. Notice, however, that $\exists y\, y' = \mathbf{0}$ is true whenever $\forall x\, \exists y\, y' = x$ is true, no matter what interpretation is given to the arithmetical symbols. Therefore the row of the truth table in which $\forall x\, \exists y\, y' = x$ is assigned the truth value 1 and $\exists y\, y' = \mathbf{0}$ is assigned the truth value 0 can never occur.

Despite this, a truth table for a formula should include all combinations of truth values, even if some cannot occur for this particular formula. This is because truth tables are intended to provide the truth value of a formula purely on the basis of how its subformulas are combined using logical connectives; they do not take into account the internal structure of the subformulas. In this way, the same truth table can be used for a variety of different formulas provided they are all constructed from subformulas in the same way using logical connectives.

In fact, truth tables and tautologies form part of the mathematical topic of logic called *propositional logic*, which considers formulas of the kind $(p \rightarrow (q \vee \neg r))$ where p, q, r are regarded as symbols which can be true or false, but which have no internal structure. In this course, because of our focus on Gödel's Theorems, we have chosen not to single out propositional logic as a subject of study by itself.

3.2 Interpretation of the quantifiers and the arithmetical symbols

Our stated intention in Subsection 1.1 was to devise a formal language adequate for number theory. With this in mind we included among the basic symbols of our language the arithmetical symbols $\mathbf{0}\ ' + \cdot$ with the intention that they would be interpreted as referring to the number 0, the successor operation ('add one'), addition and multiplication, respectively. In this subsection we shall be more specific about what the interpretation of these symbols entails and about the truth or falsity of formulas under such interpretations.

We have already noted that, even when we have fixed the interpretation of the arithmetical symbols, whether a given formula is interpreted as being true will depend on the underlying set of numbers. For example $\forall x\, \exists y\, y' = x$ is only true in a number system in which each number is the successor of some number. So it is true when interpreted as being about the integers $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$, but false when interpreted as being about the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ since in this set 0 is not the successor of any number.

We see from this that the first ingredient of an interpretation has to be a *domain* of discourse, that is a set of objects to which the formulas, or, to be more precise, to which the quantifiers in our formulas refer. Although our intention is that our language is to be interpreted as being about numbers, we shall allow for other possibilities, and hence all we stipulate about the domain of an interpretation is that it should be a non-empty set. If the domain of a given interpretation is the set U, then the universal quantifier $\forall x$ is interpreted as expressing 'for all x in U' and the existential quantifier $\exists x$ as 'for at least one x in U' or 'there is an x in U' or 'there exists an x in U'.

We also need to specify that the identity symbol $=$ should be interpreted as representing equality between elements of the domain and that the variables $x_0, x_1, x_2, \ldots$ represent unknown elements of the domain. Furthermore, the symbols (and) should be interpreted solely as punctuation marks.

Although our intention was that the arithmetical symbols $\mathbf{0}\ ' + \cdot$ should be interpreted as explained above, there is no necessity that they should be. Indeed, if the domain U does not include the number 0 then in that domain we cannot interpret the symbol $\mathbf{0}$ as referring to that number. Instead we would need to interpret it as referring to some element of the domain. In general, the symbol $\mathbf{0}$ needs to be interpreted as some specific element of the domain, but we are free to choose which this element should be.

In a similar way, although it was our intention that the symbol $'$ should be interpreted as referring to the successor operation, that is to the function $n \longmapsto n + 1$, all that is necessary is that $'$ should be interpreted as some unary operation on the elements of the domain, that is as some function $U \longrightarrow U$. Likewise the symbols $+$ and $\cdot$ each need to be interpreted as referring to binary operations on the elements of the domain, that is as functions $U \times U \longrightarrow U$.

In summary, we have the following definition.

Definition 3.2 Interpretation

An *interpretation* of the formal language consists of a non-empty set U, called the *domain* of the interpretation, along with the following:

- a specific element of U to interpret the symbol $\mathbf{0}$;
- a specific function $U \longrightarrow U$ to interpret the symbol $'$;
- specific functions $U \times U \longrightarrow U$ to interpret the symbols $+$ and $\cdot$.

Thus there are lots of possible interpretations of our formal language, but the intended one is especially important, and we single it out in the following definition.

Definition 3.3 Standard interpretation

The *standard interpretation*, denoted by $\mathcal{N}$, of the formal language has as its domain the set $\mathbb{N}$ of natural numbers. The symbol $\mathbf{0}$ is interpreted as the number 0, the symbol $'$ is interpreted as the successor operation ('add one') and the symbols $+$ and $\cdot$ are interpreted as the operations of addition and multiplication of natural numbers, respectively.

It is easily seen that in the standard interpretation the following formulas are all true:

$$\forall x \forall y\,(x \cdot y) = (y \cdot x) \qquad \exists x\,(x \cdot x) = \mathbf{0''''} \qquad \neg\exists x\, x' = \mathbf{0}$$

and the following are all false:

$$\forall x \exists y\, y' = x \qquad \exists x\,(x \cdot x) = \mathbf{0''} \qquad \forall x \exists y\,(x + y) = \mathbf{0}$$

Our formal language was devised with the standard interpretation in mind. Leibniz's Question about the existence of an algorithm for deciding which statements of number theory are true can now be reformulated in a more precise way as:

> Is there an algorithm for deciding which formulas of our formal language are true under the standard interpretation $\mathcal{N}$?

When we add to this our characterization of algorithms in terms of recursive functions, and our method for assigning Gödel numbers to formulas, this becomes the following precise mathematical question:

> Is the set of Gödel numbers of those formulas which are true in the standard interpretation $\mathcal{N}$ a recursive set?

We shall eventually answer this question in *Unit 8*.

Problem 3.4

Determine which of the following formulas are true and which are false under the standard interpretation $\mathcal{N}$.

(a) $(\mathbf{0''} + \mathbf{0'''}) = \mathbf{0''''}$

(b) $(\mathbf{0''} \cdot \mathbf{0'''}) = \mathbf{0''''}$

(c) $\exists x \forall y\,(x + y) = y$

(d) $\forall x \exists y\,((y \cdot y) + y) = x$

(e) $\forall x\,(\exists y\, x = ((\mathbf{0''} \cdot y) + \mathbf{0'}) \rightarrow \exists y\,(x \cdot x) = ((\mathbf{0''} \cdot y) + \mathbf{0'}))$

Any interpretation of the formal language which is not the standard interpretation is called a *non-standard interpretation*. From the point of view of Leibniz's Question, there may not seem to be much relevance in studying non-standard interpretations. However, we are interested in the distinction between *logical truths*, that is, formulas whose truth derives solely from their logical structure, and those formulas which express truths of number theory, that is, are true in the standard interpretation but are not logical truths. Non-standard interpretations help us to make this distinction.

As we have already noted, we can obtain non-standard interpretations simply by changing the domain to a different set of numbers. For example, we have non-standard interpretations in which the domain is the set $\mathbb{Z}$ of integers, or the set $\mathbb{R}$ of real numbers, but in which the arithmetical symbols are interpreted in the standard way for these domains. There are also

non-standard interpretations in which these symbols receive non-standard interpretations, and we now describe some of these.

Example 3.4

We take as the domain the set M of all 2×2 matrices whose entries are natural numbers.

We interpret the symbol $\mathbf{0}$ as the matrix $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$.

The interpretation of the symbol $'$ is the function $M \longrightarrow M$ given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \longmapsto \begin{pmatrix} a+1 & b \\ c & d+1 \end{pmatrix}.$$

The symbols $+$ and $\cdot$ are interpreted as the standard operations of addition and multiplication of matrices. For example, with this interpretation the formula $\forall x \forall y\,(x \cdot y) = (y \cdot x)$ is interpreted as saying that matrix multiplication in M is commutative, and hence is false. ♦

Example 3.4 illustrates that we can have non-standard interpretations where the domains are not standard sets of numbers. The next two examples are also of this kind. They will be of great importance later in the course.

Example 3.5

We take as domain the set

$$\mathbb{N}^* = \mathbb{N} \cup \{\omega\},$$

where $\mathbb{N}$ is the set of natural numbers and ω is an element not in $\mathbb{N}$. We don't need to know exactly what the element ω is; all we need to specify is which functions on $\mathbb{N}^*$ serve as the interpretations of $' + \cdot$ and which element of $\mathbb{N}^*$ serves as the interpretation of $\mathbf{0}$.

We shall denote this interpretation by $\mathcal{N}^*$.

We interpret $\mathbf{0}$ as the number 0. The interpretations of $' + \cdot$ are the functions given by the following tables.

x	0	1	2	...	n	...	ω
x'	1	2	3	...	$n+1$	...	ω

		b						
	$+$	0	1	2	...	n	...	ω
	0	0	1	2	...	n	...	ω
	1	1	2	3	...	$1+n$	...	ω
	2	2	3	4	...	$2+n$	...	ω
a	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
	m	m	$m+1$	$m+2$	...	$m+n$	...	ω
	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
	ω	ω	ω	ω	...	ω	...	ω

The values of $a + b$ and $a \cdot b$ for $a, b \in \mathbb{N}^*$ are found by looking at the entries in the appropriate a-row and b-column of these tables.

		b						
	$\cdot$	0	1	2	...	n	...	ω
	0	0	0	0	...	0	...	0
	1	0	1	2	...	n	...	ω
	2	0	2	4	...	$2n$	...	ω
a	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
	m	0	m	$2m$	...	mn	...	ω
	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
	ω	0	ω	ω	...	ω	...	ω

♦

Example 3.6

We take as the domain of this interpretation the set

$$\mathbb{N}^{**} = \mathbb{N} \cup \{\alpha, \beta\},$$

where α, β are distinct elements not in $\mathbb{N}$. The symbol $\mathbf{0}$ is interpreted as the number 0, and the functions which are the interpretations of the symbols $' + \cdot$ are given by the following tables.

We shall denote this interpretation by $\mathcal{N}^{**}$.

x	0	1	2	...	n	...	α	β
x'	1	2	3	...	$n+1$	...	α	β

The values of $a + b$ and $a \cdot b$ for $a, b \in \mathbb{N}^{**}$ are found by looking at the entries in the appropriate a-row and b-column of these tables.

		b							
	$+$	0	1	2	...	n	...	α	β
	0	0	1	2	...	n	...	β	α
	1	1	2	3	...	$1+n$	...	β	α
	2	2	3	4	...	$2+n$	...	β	α
a	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	m	m	$m+1$	$m+2$		$m+n$		β	α
	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	α	α	α	α	...	α	...	β	α
	β	β	β	β	...	β	...	β	α

		b							
	$\cdot$	0	1	2	...	n	...	α	β
	0	0	0	0	...	0	...	α	β
	1	0	1	2	...	n	...	α	β
	2	0	2	4	...	$2n$	...	α	β
a	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	m	0	m	$2m$		mn	...	α	β
	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	α	0	β	β	...	β	...	β	β
	β	0	α	α	...	α	...	α	α

♦

Now we consider some formulas that are true in the standard interpretation and ask if they are true in the non-standard interpretations of Examples 3.4, 3.5 and 3.6.

Example 3.7

Consider the following formulas.

(a) $\forall x \forall y\,(x + y) = (y + x)$

(b) $\forall x\,(\neg x = \mathbf{0} \rightarrow \exists y\, x = y')$

These are certainly true in the standard interpretation $\mathcal{N}$ since, in this interpretation, the first formula says that addition of natural numbers is commutative and the second that every natural number other than 0 is the successor of a natural number.

In the interpretation of Example 3.4, the first of these formulas is again true, since, in this interpretation, it says that addition of matrices in M is commutative. However, the second formula is not true in this interpretation.

For example the matrix $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ in M is different from the matrix $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, which is the interpretation of $\mathbf{0}$, but it cannot be obtained from any other matrix in M using the function $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \longmapsto \begin{pmatrix} a+1 & b \\ c & d+1 \end{pmatrix}$ as there do not exist natural numbers a, b, c, d such that $\begin{pmatrix} a+1 & b \\ c & d+1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$.

It can be seen from the addition table of the interpretation $\mathscr{N}^*$ of Example 3.5 that the first formula is again true, as the addition operation is commutative. The second formula is also true because, in this interpretation, every non-zero natural number is the successor of a natural number and $\omega = \omega'$.

When it comes to the interpretation $\mathscr{N}^{**}$ of Example 3.6, it can be seen from the addition table that $\alpha + \beta = \alpha$ whereas $\beta + \alpha = \beta$. Since $\alpha \neq \beta$, it follows that the addition operation is not commutative, and hence the first formula is false in this interpretation. However, the second formula is again true, as every natural number other than 0 is the successor of a natural number, $\alpha = \alpha'$ and $\beta = \beta'$. ♦

Problem 3.5

For each of the following formulas, which are true in the standard interpretation, decide whether or not it is true in each of the interpretations of Examples 3.4, 3.5 and 3.6.

(a) $\forall x \neg x' = x$

(b) $\forall x \forall y (x + y') = (x + y)'$

(c) $\forall x (\mathbf{0} \cdot x) = \mathbf{0}$.

(d) $\forall x \exists y ((y + y) = x \vee (y + y)' = x)$

(e) $\exists y \forall x (x + y) = (y + x)$

There is one aspect of interpretations which we have ignored up to now but which we need to mention. Consider the question of the truth or falsity of the following formulas in the standard interpretation $\mathscr{N}$:

$$\exists x (x \cdot x) = \mathbf{0''''} \qquad \forall y \exists x (x \cdot x) = y \qquad \exists x (x \cdot x) = y$$

The first of these formulas is true, as the equation $n \times n = 4$ has a solution $n \in \mathbb{N}$. The second formula is false, as not every natural number is the square of a natural number. But what about the third formula? The problem is that it contains a variable, namely y, which is not associated with any quantifier. Hence whether or not the formula is true depends on which element of the domain, that is, which natural number, this variable is interpreted as referring to. The formula is true in $\mathscr{N}$ if and only if the value assigned to y is a natural number which is a perfect square. Of course, this is a familiar situation in mathematics. For example, the equation $x^2 + 6 = 5x$ is true if and only if x takes one of the values 2 or 3, and the formula $\cos n\pi = 1$ is true if and only if n is an even integer.

We shall need to be precise about how we determine which variables in a formula, if any, are not associated with quantifiers. We discuss this in detail in *Unit 5*. Meanwhile we need to keep in mind that when we talk about the truth of a formula under a given interpretation this has to involve assigning values to any variables not associated with quantifiers.

3.3 Logical consequences

In this subsection we shall make more precise the idea of when one statement can be inferred from others, which will be a major topic of investigation in the remaining units. Of course we are only interested in inferences which would commonly be accepted as valid. Logically valid inferences are those which do not depend on any special interpretation of the arithmetical symbols, but whose validity derives solely from the meanings given to the logical symbols. The logician Gottlob Frege described the use of logic as 'a way that, disregarding the particular characteristics of objects, depends solely on those laws on which all knowledge rests'. The laws to which Frege refers are those that tell us which deductions are logically valid. A deduction is *logically valid* if the truth of the premises of the deduction guarantees the truth of the conclusion. We make this precise, in terms of the concept of interpretations of our formal language, as follows.

Definition 3.4 Logical consequence

The formula ψ is a *logical consequence* of the formulas $\phi_1, \phi_2, \ldots, \phi_k$ if ψ is true in every interpretation in which $\phi_1, \phi_2, \ldots, \phi_k$ are true.

Thus a deduction given by a formula ψ, based on premises given by formulas $\phi_1, \phi_2, \ldots, \phi_k$, is *logically valid* if ψ is a logical consequence of $\phi_1, \phi_2, \ldots, \phi_k$.

We can also define a *logically valid formula* as one that is true under all interpretations. Thus any tautology is a logically valid formula. However, there are formulas, such as $\forall x\, x = x$, which are logically valid but are not tautologies.

If a variable not associated with a quantifier occurs in any of the formulas $\phi_1, \phi_2, \ldots, \phi_k$ and ψ then, as was noted at the end of the previous subsection, we need to assign to it a value taken from the domain of the interpretation. If the variable occurs more than once in this way, then each such occurrence must be assigned the same value. Since the definition of logical consequence involves a reference to 'every interpretation', in applying the definition in such cases we need to consider all possible ways of assigning values to such variables.

As a very simple example of logical consequence, it is immediate from the definition that each formula ϕ_i is a logical consequence of $\phi_1, \phi_2, \ldots, \phi_k$.

Problem 3.6

Suppose that ψ is a logical consequence of ϕ and that θ is a logical consequence of ψ. Show that θ is a logical consequence of ϕ.

In general, the formula ψ will be a logical consequence of the formulas $\phi_1, \phi_2, \ldots, \phi_k$ because of the meanings that we give to the logical symbols. In some cases only the meanings of the connectives are involved. Because of the importance of such cases when we come to discuss formal proofs, we use special terminology for them.

Definition 3.5 Tautological consequence

The formula ψ is a *tautological consequence* of the formulas $\phi_1, \phi_2, \cdots, \phi_k$ if the formula

$$((\cdots((\phi_1 \,\&\, \phi_2) \,\&\, \phi_3) \cdots \,\&\, \phi_k) \rightarrow \psi)$$

is a tautology.

For $k = 1$, the formula in this definition is $(\phi_1 \rightarrow \psi)$; for $k = 2$, it is $((\phi_1 \,\&\, \phi_2) \rightarrow \psi)$, for $k = 3$, it is $(((\phi_1 \,\&\, \phi_2) \,\&\, \phi_3) \rightarrow \psi)$; and so on.

Example 3.8

Let ϕ_1 be the formula $(\exists x\,(x + y) = \mathbf{0} \to y = \mathbf{0}')$, let ϕ_2 be the formula $(\exists x\,(x + y) = \mathbf{0} \vee \forall z\, y = z)$, and let ψ be the formula $(y = \mathbf{0}' \vee \forall z\, y = z)$. It can be seen from Solution 3.2(b) that the formula $((\phi_1 \,\&\, \phi_2) \to \psi)$ is a tautology, so that ψ is a tautological consequence of the formulas ϕ_1, ϕ_2. ♦

In the next unit we introduce a rule of proof which enables us to infer ψ from the formulas $\phi_1, \phi_2, \ldots, \phi_k$ when ψ is a tautological consequence of $\phi_1, \phi_2, \ldots, \phi_k$. We wish to know that this is a logically valid deduction. The following theorem supplies that justification.

Theorem 3.1

If the formula ψ is a tautological consequence of the formulas $\phi_1, \phi_2, \ldots, \phi_k$, then ψ is a logical consequence of these formulas.

Proof

Suppose that ψ is a tautological consequence of the formulas $\phi_1, \phi_2, \ldots, \phi_k$. This means that the formula

$$((\cdots((\phi_1 \,\&\, \phi_2) \,\&\, \phi_3) \cdots \,\&\, \phi_k) \to \psi) \tag{3.3}$$

is a tautology.

Now suppose that we have an interpretation in which all the formulas $\phi_1, \phi_2, \ldots, \phi_k$ are true. From the truth table for conjunction, we see that the formulas ϕ_1, $(\phi_1 \,\&\, \phi_2)$, $((\phi_1 \,\&\, \phi_2) \,\&\, \phi_3), \ldots, ((\cdots(\phi_1 \,\&\, \phi_2) \,\&\, \phi_3) \cdots \,\&\, \phi_k)$ are all true in this interpretation. Since the formula (3.3) is a tautology, it is also true in this interpretation. Hence from the truth table for implication we can deduce that ψ is also true in this interpretation. (Otherwise, one row of the truth table would be

$$\begin{array}{l} ((((\phi_1 \,\&\, \phi_2) \,\&\, \phi_3) \cdots \,\&\, \phi_k) \to \psi \\ \qquad\qquad\quad 1 \qquad\quad 0\ \ 0 \end{array}$$

contradicting the statement that (3.3) is a tautology.)

Thus we have shown that ψ is true in every interpretation in which $\phi_1, \phi_2, \ldots, \phi_k$ are true. Hence ψ is a logical consequence of $\phi_1, \phi_2, \ldots, \phi_k$. ■

Problem 3.7

Show that if ϕ, ψ are formulas then ψ is a logical consequence of the formulas $\phi, (\phi \to \psi)$.

Logical consequence is a much broader concept than tautological consequence, although the latter will be very useful to us. For instance, the formula

$$(\mathbf{0}' + \mathbf{0}) = \mathbf{0}'$$

is a logical consequence of the formula

$$\forall x\,(x + \mathbf{0}) = x.$$

This is because, in any interpretation in which $\forall x\,(x + \mathbf{0}) = x$ is true, the formula $(x + \mathbf{0}) = x$ will be true for any value, in the domain of the interpretation, given to x. In particular, it is true for whatever value the interpretation gives to the term $\mathbf{0}'$. However, $(\mathbf{0}' + \mathbf{0}) = \mathbf{0}'$ is not a tautological consequence of $\forall x\,(x + \mathbf{0}) = x$, as

$$(\forall x\,(x + \mathbf{0}) = x \to (\mathbf{0}' + \mathbf{0}) = \mathbf{0}')$$

is not a tautology.

Problem 3.8

Show that $y' = x$ is a logical consequence of $x = y'$.

In the next unit we shall look at formal proofs involving the formulas of our language. Logical consequence will provide us with a yardstick of whether a proof is logically valid, in that we shall consider there to be a formal proof of ψ from assumptions $\phi_1, \phi_2, \ldots, \phi_k$ exactly when ψ is a logical consequence of $\phi_1, \phi_2, \ldots, \phi_k$.

SUMMARY

The unit began by introducing a *formal language* for number theory. This consists of a set of *basic symbols* together with rules which specify which finite strings of basic symbols form the *terms* and *formulas* of the language.

The language was designed to enable us to express statements of number theory. The basic symbols and the rules were chosen with this in mind. However, the rules are purely syntactical in character, making no reference to the intended meanings of the formulas. This enabled us to show that there is an algorithm for deciding which strings of basic symbols are formulas.

We also showed how we can assign numbers, called *Gödel numbers*, to strings of symbols in such a way that the algorithm mentioned in the previous paragraph can be related to the notion of a recursive function.

We next described the interpretation of our formal language. The interpretation of the connectives is given by their *truth tables* and we showed how these may be used to calculate the truth tables of formulas. Formulas that are assigned the truth value 'true', which we represent by the number 1, in every row of their truth tables are called tautologies.

To give an *interpretation* of a formula we need to specify a *domain* for the interpretation and how the arithmetical symbols are to be interpreted with respect to that domain. If we take the domain to be the set of natural numbers and we interpret the arithmetical symbols in the natural way, then we obtain what we called the *standard interpretation* $\mathcal{N}$. We drew attention to the fact that there are plenty of alternative, *non-standard* interpretations.

We need the idea of a non-standard interpretation so that we can distinguish between logically valid formulas and those which may be true under the standard interpretation yet false in other interpretations. We defined *logical consequence* in terms of the truth of formulas in arbitrary interpretations. Finally we considered one particular case, called *tautological consequence*, which arises when the validity of an inference depends only on the interpretation of the connectives.

OBJECTIVES

We list those topics on which we may set assessment questions to test your understanding of this unit.

After working through the unit you should be able to:

(a) recall the definitions of the *basic symbols*, a *term*, an *atomic formula* and a *formula* of our formal language;

(b) determine whether a given string of basic symbols is a term, an atomic formula or a formula of our formal language;

(c) find all the subformulas of a given formula;

(d) determine the Gödel number of a string of basic symbols (in prime power form), and determine whether a natural number given in prime power form is the Gödel number of a string of basic symbols;

(e) write out the truth table of a formula and determine whether a given formula is a tautology;

(f) in simple cases, given a formula and an interpretation, determine whether the formula is true in the given interpretation;

(g) determine whether a given formula is a tautological consequence of a given set of formulas.

ADDITIONAL EXERCISES

Most of these exercises provide further practice, should you feel you need it, in handling the main ideas in the unit on which you are likely to be assessed.

Section 1

1 Decide which of the following strings of basic symbols are terms.

(a) $x_0 \cdot (x_0 \cdot (\mathbf{0}' + \mathbf{0}))'$

(b) $((x_0 \cdot \mathbf{0}') + (x_0 \cdot \mathbf{0}))$

(c) $(x_3 = ((x_1 + x_2) \cdot \mathbf{0}''))$

Section 2

1 For each of the following strings of symbols, decide whether or not the string is an atomic formula.

(a) $x_1 + \mathbf{0} = (\mathbf{0} + x_1)$

(b) $(x_0 \cdot x_1') = ((x_0 \cdot x_1) + x_0)$

(c) $\forall x_0\,(x_0 + x_1') = (x_0 + x_1)'$

2 For each of the following strings of symbols, decide whether or not the string is a formula. If it is not a formula, explain why.

(a) $(\forall x_0\, x_0 = \mathbf{0}' \vee x_0 = \mathbf{0}')$

(b) $\forall x_2\,((x_0 \cdot x_1) = x_2 \rightarrow \exists x_2\,(x_0 \cdot x_1) = (x_0 \cdot x_1 \cdot x_1))$

(c) $\exists x_1\,\forall x_2\,(x_0 \cdot (x_2 + (x_1 + \mathbf{0}')))$

3 List the subformulas of the formula

$$(\forall x_1\,(x_0 = x_1 \vee x_0 = \mathbf{0}) \,\&\, \exists x_2\, x_0 = x_2)$$

Find the Gödel number of each of these subformulas.

4 For each of the following numbers, decide whether or not it is the Gödel number of a formula.

(a) $2^8 3^{15} 5^9 7^{16} 11^{15} 13^{14} 17^1 19^{16} 23^{12} 29^{16} 31^{12} 37^{16} 41^2$

(b) $2^1 3^{15} 5^{14} 7^{16} 11^5 13^{15} 17^{11} 19^{14} 23^{16} 29^{11} 31^2$

(c) $2^1 3^{15} 5^{14} 7^{16} 11^5 17^{11} 19^{14}$

Section 3

1 Show that each of the following formulas is a tautology.

(a) $((\neg\forall z\,(z = (y + x) \,\&\, y = \mathbf{0}) \to \exists x\, x = z) \vee (\neg\exists x\, x = z \vee \forall z\,(z = (y + x) \,\&\, y = \mathbf{0})))$

(b) $(((x = y \vee \exists x\, x = y) \to \forall x\,(x = y \vee \exists x\, x = y)) \to (\neg\forall x\,(x = y \vee \exists x\, x = y) \to \neg\, x = y))$

2 Determine the truth table for the formula $((\phi \to \psi) \,\&\, (\psi \to \phi))$ and show that it matches the truth table for $(\phi \leftrightarrow \psi)$.

3 (a) Work out the truth tables for the formulas

$$((\phi \,\&\, \psi) \vee \chi) \quad \text{and} \quad (\phi \,\&\, (\psi \vee \chi)).$$

(b) Is the formula

$$(((\phi \,\&\, \psi) \vee \chi) \leftrightarrow (\phi \,\&\, (\psi \vee \chi)))$$

a tautology? Explain your answer.

4 Show that there is an interpretation of the formal language for which the domain is a set with one element.

5 For each of the following formulas, which are true in the standard interpretation, decide whether or not it is true in each of the interpretations of Examples 3.4, 3.5 and 3.6.

(a) $\forall x \forall y\,(x \cdot y) = (y \cdot x)$

(b) $\forall x \forall y\,(x \cdot y') = ((x \cdot y) + x)$

(c) $\forall x\,(\mathbf{0} + x) = x$

6 Consider the following two non-standard interpretations of the formal language.

(1) The domain is the set $\mathbb{Z}$ of integers, with $+$ interpreted as the usual addition of integers, $\cdot$ interpreted as the usual subtraction of integers, $'$ interpreted as the function $\mathbb{Z} \longrightarrow \mathbb{Z}$ given by $n \longmapsto -n$ and the symbol $\mathbf{0}$ interpreted as the integer 3.

(2) The domain is the set $\mathbb{R}$ of real numbers, with $+$ and $\cdot$ interpreted as the usual addition and multiplication of real numbers, $'$ interpreted as the function $\mathbb{R} \longrightarrow \mathbb{R}$ given by $r \longmapsto r^2$ and the symbol $\mathbf{0}$ interpreted as the real number -1.

For each of the following formulas and each of these interpretations, decide whether or not the formula is true in the interpretation.

(a) $\forall x \forall y \forall z\,((x + y) \cdot z) = (x + (y \cdot z))$

(b) $\forall x \forall y \forall z\,((x + y) \cdot z) = ((x \cdot z) + (y \cdot z))$

(c) $\forall x \forall y \exists z\,(x + z') = y$

(d) $\forall x \exists y\,(y' = x \vee (\mathbf{0} \cdot y') = x)$

7 Show that the formula

$$(\forall x\, x = \mathbf{0} \vee \neg\exists x\, x = \mathbf{0})$$

is a tautological consequence of the formulas

$$\neg\forall x\, x = \mathbf{0} \quad \text{and} \quad \neg(\exists x(x = \mathbf{0} \leftrightarrow \forall x\, x = \mathbf{0}) \rightarrow \exists x\, x = \mathbf{0})$$

SOLUTIONS TO THE PROBLEMS

Solution 1.1

The statement 'x_0 is even' can be written as $\exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1)$. Thus the statement 'the square of every even number is even' can be written as

$$\forall x_0\, (\exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1) \rightarrow \exists x_1\, (x_0 \cdot x_0) = (\mathbf{0}'' \cdot x_1))$$

Alternative formulations are possible. For example, one alternative formulation, which you may find clearer, is

$$\forall x_0\, (\exists x_1\, x_0 = (\mathbf{0}'' \cdot x_1) \rightarrow \exists x_2\, (x_0 \cdot x_0) = (\mathbf{0}'' \cdot x_2))$$

Solution 1.2

(a) (i) The sequence meets the requirements of the definition.

(ii) The sequence does not meet the requirements of the definition. The term $(x_1 + \mathbf{0}'')$ which occurs third in the sequence is not of the form $(\tau_i + \tau_j)$ where τ_i, τ_j are earlier terms in the sequence.

(iii) The sequence meets the requirements of the definition. (Note that the second appearance of the term x_1 in the sequence is redundant, although it could be argued that it makes the final production of the term $((x_1 \cdot \mathbf{0}')' + x_1)$ easier to follow. So Definition 1.2 hasn't completely avoided redundancy.)

(b) In each case the final string is a term. In cases (i) and (iii) this follows immediately from the fact that the sequences comply with the requirements of Definition 1.2. In case (ii), although the given sequence does not comply with this definition, the alternative sequence

$$x_1,\ x_2,\ \mathbf{0},\ \mathbf{0}',\ \mathbf{0}'',\ (x_1 + \mathbf{0}''),\ (x_2 \cdot (x_1 + \mathbf{0}''))$$

does comply with the definition, and this shows that $(x_2 \cdot (x_1 + \mathbf{0}''))$ is a term.

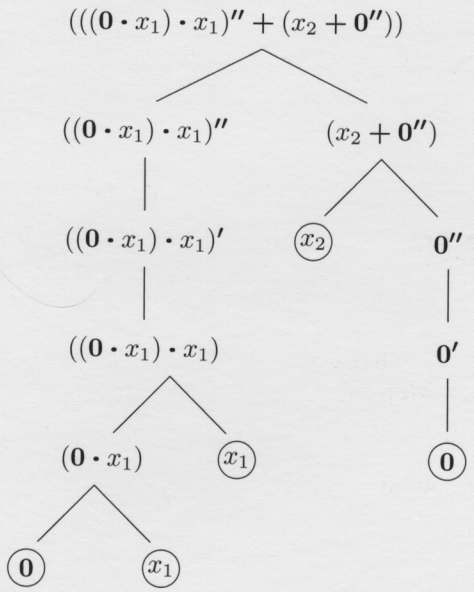
Solution 1.3

One solution is the sequence

$$\mathbf{0},\ x_0,\ x_2,\ x_3,\ \mathbf{0}',\ (x_0 + \mathbf{0}'),\ (x_0 + \mathbf{0}')',\ x_2',\ \mathbf{0}'',\ ((x_0 + \mathbf{0}')' \cdot x_2'),$$
$$(x_3 + \mathbf{0}''),\ (((x_0 + \mathbf{0}')' \cdot x_2') \cdot (x_3 + \mathbf{0}'')),\ (((x_0 + \mathbf{0}')' \cdot x_2') \cdot (x_3 + \mathbf{0}''))'.$$

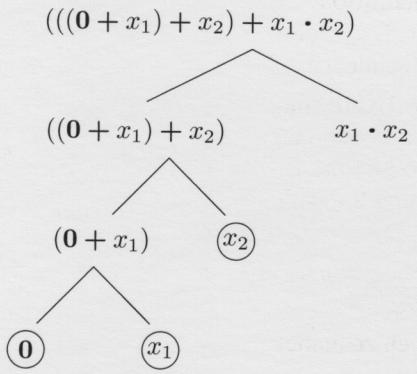
Solution 1.4

(a) The diagram is as follows.



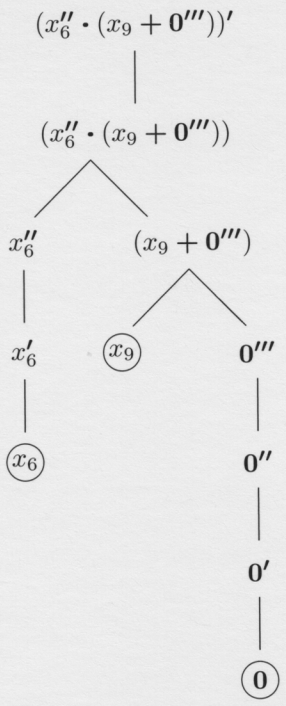
Hence the original string is a term.

(b) The diagram is as follows.



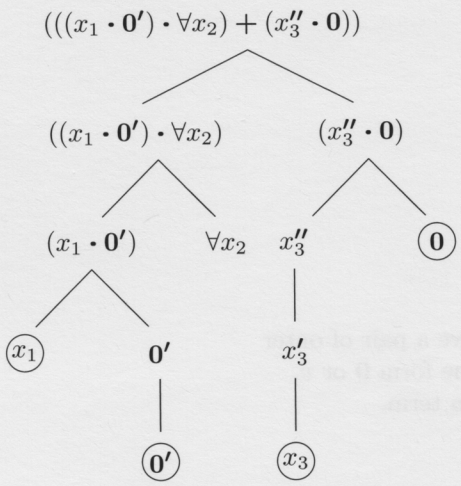
The string $x_1 \cdot x_2$ doesn't end with a $'$ and doesn't have a pair of outer brackets, and so is a terminal string; but it is not of the form $\mathbf{0}$ or v, where v is a variable. Hence the original string is not a term.

(c) The diagram is as follows.



Hence the original string is a term.

(d) The diagram is as follows.



The string $\forall x_2$ doesn't end with a $'$ and doesn't have a pair of outer brackets, and so is a terminal string; but it is not of the form $\mathbf{0}$ or v, where v is a variable. Hence the original string is not a term. (Of course, this analysis is not really necessary since no term can include the symbol $\forall$.)

Solution 2.1

(a) As $(x_1 + x_2)'$ and $(x_1 + x_2')$ are both terms, this is an atomic formula.

(b) Similarly, this is an atomic formula.

(c) This involves more than one $=$ sign, so is not an atomic formula.

(d) A blank space is not a term, so this is not an atomic formula.

(e) Neither $x_0 \cdot \mathbf{0}$ nor $\mathbf{0} \cdot x_0$ is a term, as outer brackets are missing, so this is not an atomic formula.

Solution 2.2

(a) The bracket count is as follows.

$$\begin{array}{ccccccccc} (x_0 = \mathbf{0'} \vee (x_0 = (\mathbf{0'} \cdot x_1) \rightarrow x_0 = (x_1 \cdot x_1))) \\ 1 \quad 2 \quad 3 \quad 2 \quad 3 \quad 210 \end{array}$$

From this we see that the first bracket is associated with the number 1 and the first a-symbol following it is a connective, namely $\vee$. There is no other bracket with this property. So this string has a principal connective, namely $\vee$.

(b) The bracket count is as follows.

$$\begin{array}{ccccccccc} (\exists x_1 ((x_1 \cdot x_1) = x_0) \,\&\, x_0 = ((x_1 \cdot \mathbf{0''}) + \mathbf{0'})) \\ 1 \quad 23 \quad 2 \quad 1 \quad 23 \quad 2 \quad 10 \end{array}$$

For similar reasons we see that this string has a principal connective, namely &.

(c) The bracket count is as follows.

$$\begin{array}{ccccccccccccc} (((x_0 + x_0) + x_1') = (x_2 + x_0) \vee (x_0 \cdot x_0) = x_1 \rightarrow \neg \exists x_2 (x_2 \cdot x_1) = x_0) \\ 123 \quad 2 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 0 \end{array}$$

Here there are two brackets associated with the number 1 such that the first a-symbol following them is a connective. Hence this string does not have a principal connective. (It follows that it is not a formula.)

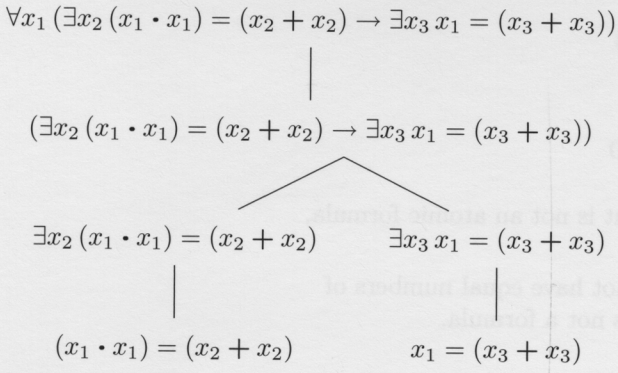
(d) The bracket count is as follows.

$$\begin{array}{ccccccccc} (\exists x_0 ((x_1 = (x_0 \cdot x_2)) \rightarrow x_1 = (x_2 \cdot x_2))) \\ 1 \quad 23 \quad 4 \quad 32 \quad 3 \quad 210 \end{array}$$

This string does not have a bracket associated with the number 1 such that the next a-symbol following it is a connective. So this string does not have a principal connective. (Again, it follows that it is not a formula.)

Solution 2.3

(a) As ever, a bracket count helps to identify the principal connective. The diagram is as follows.



The terminal strings are atomic formulas, so the original string is a formula.

(b) The diagram is as follows.

$$\forall x_0\, \neg \exists x_1\, (x_1 \cdot x_1) = x_0$$
$$\neg \exists x_1\, (x_1 \cdot x_1) = x_0$$
$$\exists x_1\, (x_1 \cdot x_1) = x_0$$
$$(x_1 \cdot x_1) = x_0$$

The terminal string is an atomic formula, so the original string is a formula.

(c) The diagram is as follows.

$$((\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \vee x_1 = \mathbf{0}) \vee x_1 = \mathbf{0'})$$
$$(\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \vee x_1 = \mathbf{0}) \qquad x_1 = \mathbf{0'}$$
$$\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \qquad x_1 = \mathbf{0}$$
$$x_1 = (x_0 + \mathbf{0''})$$

The terminal strings are atomic formulas, so the original string is a formula.

(d) The diagram is as follows.

$$(x_0 = \mathbf{0} \vee x_1 = \mathbf{0}\ \&\ x_2 = 0)$$
$$x_0 = \mathbf{0} \qquad x_1 = \mathbf{0}\ \&\ x_2 = 0$$

The string on the right is a terminal string but is not an atomic formula, so the original string is not a formula.

(e) A bracket count shows that this string does not have equal numbers of left-hand and right-hand brackets. Hence it is not a formula.

Solution 2.4

We simply need to use a slightly different system of bracketing:

$$\forall x_0\, \exists x_1\, ((\neg\, x_1 = \mathbf{0'}\ \&\ \forall x_2\, (\exists x_3\, (x_2 \cdot x_3) = x_1 \rightarrow (x_2 = \mathbf{0'} \vee x_2 = x_1))) \,\&\, \exists x_4\, (x_0 + x_4) = x_1)$$

Solution 2.5

(a) From the analysis in Solution 2.3(b), we see that the subformulas are:

$(x_1 \cdot x_1) = x_0$

$\exists x_1\,(x_1 \cdot x_1) = x_0$

$\neg\exists x_1\,(x_1 \cdot x_1) = x_0$

$\forall x_0 \neg\exists x_1\,(x_1 \cdot x_1) = x_0$

(b) From the analysis in Solution 2.3(c), we see that the subformulas are:

$x_1 = (x_0 + \mathbf{0''})$

$x_1 = \mathbf{0}$

$x_1 = \mathbf{0'}$

$\exists x_0\, x_1 = (x_0 + \mathbf{0''})$

$(\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \vee x_1 = \mathbf{0})$

$((\exists x_0\, x_1 = (x_0 + \mathbf{0''}) \vee x_1 = \mathbf{0}) \vee x_1 = \mathbf{0'})$

(c) The formula is atomic, so is its only subformula.

Solution 2.6

(a) $2^{18}3^{14}5^{1}7^{1}11^{17}13^{13}17^{10}19^{11}23^{11}29^{2}31^{12}37^{15}41^{2}$

(b) $2^{8}3^{15}5^{1}7^{15}11^{11}13^{14}17^{10}19^{11}23^{6}29^{15}31^{14}37^{10}41^{2}$

Solution 2.7

(a) This is the Gödel number of the string $\neg\, x_4 = x_1)$.

(b) This is not the Gödel number of a string as it is divisible by 23 but not by the smaller prime 19.

(c) This is the Gödel number of the string $\forall x_0\, \exists x_1\, x_0' = \forall$.

Solution 3.1

(a)

$((\phi$	$\&$	$\psi)$	$\&$	$\chi)$
1	1	1	1	1
1	1	1	0	0
1	0	0	0	1
1	0	0	0	0
0	0	1	0	1
0	0	1	0	0
0	0	0	0	1
0	0	0	0	0
①	②	①	③	①

(b)

$((\phi$	$\rightarrow$	$\psi)$	$\rightarrow$	$\phi)$
1	1	1	1	1
1	0	0	1	1
0	1	1	0	0
0	1	0	0	0
①	②	①	③	①

(c)

$(\phi$	$\leftrightarrow$	$\phi)$
1	1	1
0	1	0
①	②	①

(d)

$(\neg$	$(\phi$	$\vee$	$\psi)$	$\leftrightarrow$	$\chi)$
0	1	1	1	0	1
0	1	1	1	1	0
0	1	1	0	0	1
0	1	1	0	1	0
0	0	1	1	0	1
0	0	1	1	1	0
1	0	0	0	1	1
1	0	0	0	0	0
③	①	②	①	④	①

Solution 3.2

(a) As we know we are dealing with a formula, we shall only give as much of the analysis as is needed to establish which subformulas to use in the truth table.

$$(\forall x\, \exists y\, y' = x \rightarrow \exists y\, y' = \mathbf{0})$$

$$\underline{\forall x\, \exists y\, y' = x} \qquad \underline{\exists y\, y' = \mathbf{0}}$$

We have underlined the subformulas to be used in the table. Here they are $\forall x\, \exists y\, y' = x$ and $\exists y\, y' = \mathbf{0}$. If we denote these by ϕ and ψ respectively, the entire formula is $(\phi \rightarrow \psi)$ and its truth table is the standard truth table for implication, namely

$(\phi$	$\rightarrow$	$\psi)$
1	1	1
1	0	0
0	1	1
0	1	0

(b) The analysis is as follows.

$$(((\exists x\,(x + y) = \mathbf{0} \rightarrow y = \mathbf{0}') \,\&\, (\exists x\,(x + y) = \mathbf{0} \vee \forall z\, y = z)) \rightarrow (y = \mathbf{0}' \vee \forall z\, y = z))$$

$$((\exists x\,(x + y) = \mathbf{0} \rightarrow y = \mathbf{0}') \,\&\, (\exists x\,(x + y) = \mathbf{0} \vee \forall z\, y = z)) \qquad (y = \mathbf{0}' \vee \forall z\, y = z)$$

$$(\exists x\,(x + y) = \mathbf{0} \rightarrow y = \mathbf{0}') \qquad (\exists x\,(x + y) = \mathbf{0} \vee \forall z\, y = z) \qquad \underline{y = \mathbf{0}'} \qquad \underline{\forall z\, y = z}$$

$$\underline{\exists x\,(x + y) = \mathbf{0}} \qquad \underline{y = \mathbf{0}'} \qquad \underline{\exists x\,(x + y) = \mathbf{0}} \qquad \underline{\forall z\, y = z}$$

We see from this analysis that there are three distinct subformulas: $\exists x\,(x + y) = \mathbf{0}$, $y = \mathbf{0}'$ and $\forall z\, y = z$. If we denote these by ϕ, ψ and χ respectively, then the entire formula is

$$(((\phi \rightarrow \psi) \,\&\, (\phi \vee \chi)) \rightarrow (\psi \vee \chi))$$

The truth table for this formula is as follows.

$(((\phi$	$\rightarrow$	$\psi)$	$\&$	$(\phi$	$\vee$	$\chi))$	$\rightarrow$	$(\psi$	$\vee$	$\chi))$
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	0
1	0	0	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	1	1	1	1
0	1	1	0	0	0	0	1	1	1	0
0	1	0	1	0	1	1	1	0	1	1
0	1	0	0	0	0	0	1	0	0	0
①	②	①	③	①	②	①	④	①	②	①

Solution 3.3

Analysis of the formula shows that it can be written as

$$(\phi \rightarrow (\neg\psi \rightarrow (\psi \rightarrow \chi)))$$

where ϕ, ψ and χ are the subformulas $\forall x\, x = y$, $\exists z\, (z + \mathbf{0''}) = y$ and $x = \mathbf{0'}$, respectively. Its truth table is as follows.

$(\phi$	$\rightarrow$	$(\neg$	ψ	$\rightarrow$	$(\psi$	$\rightarrow$	$\chi)))$
1	1	0	1	1	1	1	1
1	1	0	1	1	1	0	0
1	1	1	0	1	0	1	1
1	1	1	0	1	0	1	0
0	1	0	1	1	1	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	0	1	1
0	1	1	0	1	0	1	0
①	④	②	①	③	①	②	①

We see from column ④ of this truth table that the formula is a tautology.

Solution 3.4

In each case we first give the meaning of the formula in the standard interpretation $\mathcal{N}$, and then say whether or not it is true.

(a) 'The sum of 2 and 3 is 5.' True.

(b) 'The product of 2 and 3 is 5.' False.

(c) 'There is an $x \in \mathbb{N}$ such that, for all $y \in \mathbb{N}$, $x + y = y$.' True. We can take $x = 0$.

(d) 'Every $x \in \mathbb{N}$ can be expressed in the form $y^2 + y$ with $y \in \mathbb{N}$.' False. For example, the number 1 cannot be expressed in this form.

(e) 'For all $x \in \mathbb{N}$, if x is odd, then x^2 is odd.' True.

Solution 3.5

In each case, we first give the meaning of the formula and then say whether it is true in each of the interpretations.

(a) 'The equation $x' = x$ has no solutions.'

3.4 True. For each matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M$,

$\begin{pmatrix} a & b \\ c & d \end{pmatrix}' = \begin{pmatrix} a+1 & b \\ c & d+1 \end{pmatrix}$ and hence is different from $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

3.5 False. We have $\omega' = \omega$, so $x = \omega$ is a solution of the equation $x' = x$.

3.6 False. We have $\alpha' = \alpha$ and $\beta' = \beta$, so both $x = \alpha$ and $x = \beta$ are solutions of the equation $x' = x$.

(b) 'For all x, y in the domain, $x + y' = (x + y)'$.'

3.4 True. For all $\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \begin{pmatrix} e & f \\ g & h \end{pmatrix} \in M$,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e+1 & f \\ g & h+1 \end{pmatrix} = \begin{pmatrix} a+e+1 & b+f \\ c+g & d+h+1 \end{pmatrix}.$$

3.5 True. Whenever $x, y \in \mathbb{N}$, the equation $x + y' = (x + y)'$ is equivalent to $x + (y + 1) = (x + y) + 1$, which is true. It remains only to check the cases where $x = \omega$ or $y = \omega$. We see that in all such cases, either $x = \omega$ or $y' = \omega$ and hence $x + y' = \omega$. Similarly $x + y = \omega$ and hence $(x + y)' = \omega$. So in all these cases $x + y' = (x + y)'$. Hence this equation holds for all $x, y \in \mathbb{N}^*$.

3.6 True. This time there are more cases to check. We have that:

if $x, y \in \mathbb{N}$, then $x + y' = x + (y + 1)$ and $(x + y)' = (x + y) + 1$;
if $x \in \mathbb{N}$, $y = \alpha$, then $x + y' = x + \alpha = \beta$ and $(x + y)' = \beta' = \beta$;
if $x \in \mathbb{N}$, $y = \beta$, then $x + y' = x + \beta = \alpha$ and $(x + y)' = \alpha' = \alpha$;
if $x = \alpha$, $y \neq \alpha$, then $x + y' = \alpha$ and $(x + y)' = \alpha' = \alpha$;
if $x = y = \alpha$, then $x + y' = \alpha + \alpha' = \alpha + \alpha = \beta$ and $(x + y)' = (\alpha + \alpha)' = \beta' = \beta$;
if $x = \beta$, $y \neq \beta$, then $x + y' = \beta$ and $(x + y)' = \beta' = \beta$;
if $x = y = \beta$, then $x + y' = \beta + \beta' = \beta + \beta = \alpha$ and $(x + y)' = (\beta + \beta)' = \alpha' = \alpha$.

Hence the equation $x + y' = (x + y)'$ holds for all $x, y \in \mathbb{N}^{**}$.

(c) 'For all x in the domain, $0 \cdot x = 0$.'

3.4 True. For every matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M$,

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

3.5 True. For $x \in \mathbb{N}$, $0 \cdot x = 0$ and also $0 \cdot \omega = 0$.

3.6 False. We have $0 \cdot \alpha = \alpha$ and $0 \cdot \beta = \beta$.

(d) 'For all x there is some y such that $(y + y) = x$ or $(y + y)' = x$.'

3.4 False. For instance, take for x the matrix $\begin{pmatrix} 0 & 3 \\ 5 & 0 \end{pmatrix}$ in M. Whatever matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ we take for y, $\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 2a & 2b \\ 2c & 2d \end{pmatrix}$, which can never equal $\begin{pmatrix} 0 & 3 \\ 5 & 0 \end{pmatrix}$ for natural numbers b, c. Thus, for this x, there is no y in M for which the statement '$(y + y) = x$' is true. Likewise, for any matrix in M, $\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} a & b \\ c & d \end{pmatrix}\right)' = \begin{pmatrix} 2a+1 & 2b \\ 2c & 2d+1 \end{pmatrix}$, which can never equal $\begin{pmatrix} 0 & 3 \\ 5 & 0 \end{pmatrix}$ for natural numbers b, c. Thus, for this x, there is also no y in M for which the statement '$(y + y)' = x$' is true. Therefore there is no y in M for which the statement '$(y + y) = x$ or $(y + y)' = x$' is true.

3.5 True. For any $x \in \mathbb{N}$, x is of the form $2y$ or $2y+1$ for some $y \in \mathbb{N}$, so given such an x select the corresponding y accordingly. When x is ω we have $\omega + \omega = \omega$, so that by taking y to be ω, the statement '$(y+y) = x$' is true, and thus the statement '$(y+y) = x$ or $(y+y)' = x$' is also true.

It is also the case that $(\omega + \omega)' = \omega$, so that, when x is ω, we can take $y = \omega$ making the statement '$(y+y)' = x$' true, and thus the statement '$(y+y) = x$ or $(y+y)' = x$' true.

3.6 True. For any $x \in \mathbb{N}$, x is of the form $2y$ or $2y+1$ for some $y \in \mathbb{N}$, so given such an x select the corresponding y accordingly. When x is α, we can take y to be β, as then $\beta + \beta = \alpha$, so that '$(y+y) = x$' and thus '$(y+y) = x$ or $(y+y)' = x$' are true. When x is β, we can take y to be α, as then $\alpha + \alpha = \beta$, so that '$(y+y) = x$' and thus '$(y+y) = x$ or $(y+y)' = x$' are true.

(e) 'There is a y such that, for all x, $(x+y) = (y+x)$.'

3.4 True. Take y to be the matrix $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ in M. Then, for all matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ in M,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

so that, for this y and all x in M, we have $x + y = y + x$.

3.5 True. Take y to be 0. Then for any x in $\mathbb{N}^*$ we have $0 + x = x = x + 0$, so that for this y and all x in $\mathbb{N}^*$ we have $x + y = y + x$.

3.6 False. To demonstrate this we need to show that, for all y in $\mathbb{N}^{**}$, there is some x for which $x + y \neq y + x$. If $y \in \mathbb{N}$, we have $\alpha + y = \alpha$ and $y + \alpha = \beta \neq \alpha$. If y is α then, for any natural number x, $x + \alpha = \beta \neq \alpha = \alpha + x$. Lastly, if y is β then, for any natural number x, $x + \beta = \alpha \neq \beta = \beta + x$.

Solution 3.6

In any interpretation in which ϕ is true, ψ is true (as ψ is a logical consequence of ϕ), so that θ is also true (as θ is a logical consequence of ψ). Thus θ is true in every interpretation in which ϕ is true, so that θ is a logical consequence of ϕ.

Solution 3.7

The truth table for the formula $((\phi \,\&\, (\phi \to \psi)) \to \psi)$ is as follows.

$((\phi$	$\&$	$(\phi$	$\to$	$\psi))$	$\to$	$\psi)$
1	1	1	1	1	1	1
1	0	1	0	0	1	0
0	0	0	1	1	1	1
0	0	0	1	0	1	0

From this truth table we see that $((\phi \,\&\, (\phi \to \psi)) \to \psi)$ is a tautology. Hence ψ is a tautological consequence of $\phi, (\phi \to \psi)$. Therefore, by Theorem 3.1, it is a logical consequence of these formulas.

Solution 3.8

Take any interpretation, which can have any non-empty set as domain. Because we interpret the symbol $=$ to mean equality between elements of the domain, any way of assigning values to the variables x and y which makes $x = y'$ true automatically makes $y' = x$ true. Thus $y' = x$ is true in all interpretations in which $x = y'$ is true, so that $y' = x$ is a logical consequence of $x = y'$.

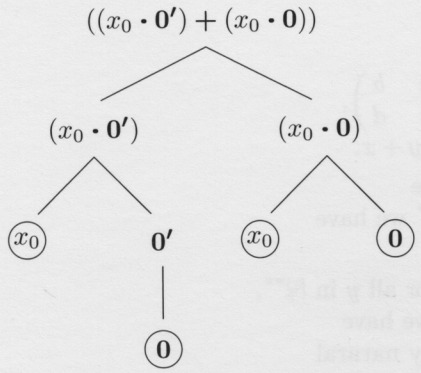
This is a further example of a logical consequence which is not a tautological consequence, as $(x = y' \to y' = x)$ is not a tautology.

SOLUTIONS TO ADDITIONAL EXERCISES

Section 1

1 (a) The string ends with a $'$, so we remove this and analyse the shorter string $x_0 \cdot (x_0 \cdot (\mathbf{0}' + \mathbf{0}))$. As this doesn't end with a $'$ and doesn't start with a left-hand bracket, it is a terminal string; but it is not of the form $\mathbf{0}$ or v, where v is a variable. So the original string is not a term.

(b) This string is a term and has the following analysis.



(c) We can see that the string involves the symbol $=$ which is not allowed in a term. So the string is not a term.

Note that our method of analysis would detect that the string is not a term as follows. As the string starts and ends with the correct sorts of bracket, we do the the bracket count, giving

$$\begin{array}{cccccc} (& x_3 & = & (& (& x_1 + x_2) \cdot \mathbf{0}'')) \\ 1 & & & 2 & 3 & 1 \quad 0 \end{array}$$

There is no $+$ or $\cdot$ in the place where the principal symbol is expected, just the symbol $=$, so the analysis tells us that the string is not a term — it is a terminal string that is not of the form $\mathbf{0}$ or v, where v is a variable.

Here we have an example of case (b) of Rule 2 for terms.

Section 2

1 (a) This is not an atomic formula because $x_1 + \mathbf{0}$ is not a term (brackets are missing).

(b) As $(x_0 \cdot x_1')$ and $((x_0 \cdot x_1) + x_0)$ are terms, this is an atomic formula.

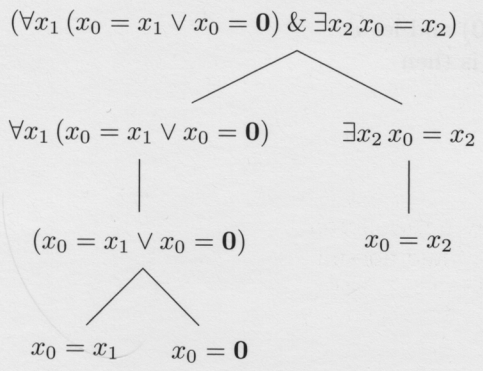
(c) As $\forall x_0\,(x_0 + x_1')$ is not a term, this is not an atomic formula (but it is a formula).

2 (a) This is a formula.

(b) This is not a formula: the string $(x_0 \cdot x_1 \cdot x_1)$ is not a term (it is not adequately bracketed).

(c) This is not a formula: the symbol $=$ does not occur so the string cannot be built up from atomic formulas in accordance with Definition 2.2.

3 The formula can be analysed as follows.



The formula has the following seven subformulas.

$x_0 = x_1$

$x_0 = \mathbf{0}$

$x_0 = x_2$

$((x_0 = x_1 \vee x_0 = \mathbf{0})$

$\forall x_1\,(x_0 = x_1 \vee x_0 = \mathbf{0})$

$\exists x_2\, x_0 = x_2$

$(\forall x_1\,(x_0 = x_1 \vee x_0 = \mathbf{0}) \,\&\, \exists x_2\, x_0 = x_2)$

The Gödel numbers of these formulas are, respectively, as follows.

$2^{15}3^{14}5^{16}$

$2^{15}3^{14}5^{10}$

$2^{15}3^{14}5^{17}$

$2^{1}3^{15}5^{14}7^{16}11^{4}13^{15}17^{14}19^{10}23^{2}$

$2^{8}3^{16}5^{1}7^{15}11^{14}13^{16}17^{4}19^{15}23^{14}29^{10}31^{2}$

$2^{9}3^{17}5^{15}7^{14}11^{17}$

$2^{1}3^{8}5^{16}7^{1}11^{15}13^{14}17^{16}19^{4}23^{15}29^{14}31^{10}37^{2}41^{3}43^{9}47^{17}53^{15}59^{14}61^{17}67^{2}$

4 (a) This is the Gödel number of the string

$$\forall x_0\, \exists x_1\, x_0 = (x_1 + x_1 + x_1)$$

This string is not a formula. The string $(x_1 + x_1 + x_1)$ is not a term and hence $x_0 = (x_1 + x_1 + x_1)$ is not an atomic formula.

(b) This is the Gödel number of the string

$$(x_0 = x_1 \to x_0' = x_1')$$

This string is a formula.

(c) This is not even the Gödel number of a string of basic symbols since it is divisible by 17 but not by the smaller prime 13. Thus it is certainly not the Gödel number of a formula.

Section 3

1 (a) Let ϕ denote the subformula $\forall z\,(z = (y + x)\ \&\ y = \mathbf{0})$ and let ψ denote the subformula $\exists x\, x = z$. The given formula is then

$$((\neg\phi \to \psi) \vee (\neg\psi \vee \phi))$$

and its truth table is as follows.

$((\neg$	ϕ	$\to$	$\psi)$	$\vee$	$(\neg$	ψ	$\vee$	$\phi))$
0	1	1	1	1	0	1	1	1
0	1	1	0	1	1	0	1	1
1	0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	1	0

↑
tautology

(b) Let ϕ denote the subformula $x = y$, let ψ denote the subformula $\exists x\, x = y$ and let χ denote the subformula $\forall x\,(x = y \vee \exists x\, x = y)$. The given formula is then

$$(((\phi \vee \psi) \to \chi) \to (\neg\chi \to \neg\phi))$$

and its truth table is as follows.

$(((\phi$	$\vee$	$\psi)$	$\to$	$\chi)$	$\to$	$(\neg$	χ	$\to$	$\neg$	$\phi))$
1	1	1	1	1	1	0	1	1	0	1
1	1	1	0	0	1	1	0	0	0	1
1	1	0	1	1	1	0	1	1	0	1
1	1	0	0	0	1	1	0	0	0	1
0	1	1	1	1	1	0	1	1	1	0
0	1	1	0	0	1	1	0	1	1	0
0	0	0	1	1	1	0	1	1	1	0
0	0	0	1	0	1	1	0	1	1	0

↑
tautology

2 Our table shows the intermediate stages in working out the truth table for $((\phi \to \psi)\ \&\ (\psi \to \phi))$ alongside the table for $(\phi \leftrightarrow \psi)$.

ϕ	ψ	$(\phi \to \psi)$	$(\psi \to \phi)$	$((\phi \to \psi)\ \&\ (\psi \to \phi))$	$(\phi \leftrightarrow \psi)$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	1	0	0	0
0	0	1	1	1	1

3 (a) The truth tables are as follows.

$((\phi$	$\&$	$\psi)$	$\vee$	$\chi)$
1	1	1	1	1
1	1	1	1	0
1	0	0	1	1
1	0	0	0	0
0	0	1	1	1
0	0	1	0	0
0	0	0	1	1
0	0	0	0	0

$(\phi$	$\&$	$(\psi$	$\vee$	$\chi))$
1	1	1	1	1
1	1	1	1	0
1	1	0	1	1
1	0	0	0	0
0	0	1	1	1
0	0	1	1	0
0	0	0	1	1
0	0	0	0	0

That these formulas do not have the same truth table is one reason why we reject the string $(\phi\ \&\ \psi \vee \chi)$ as a formula. It is an ambiguity comparable to that of the numerical expression $(1 \times 2 + 3)$.

(b) The formula $(((\phi \,\&\, \psi) \vee \chi) \leftrightarrow (\phi \,\&\, (\psi \vee \chi)))$ is not a tautology. For instance, when ϕ, ψ, χ take, respectively, the values 0, 0, 1, the formula $((\phi \,\&\, \psi) \vee \chi)$ takes the value 1 and the formula $(\phi \,\&\, (\psi \vee \chi))$ takes the value 0, so that $(((\phi \,\&\, \psi) \vee \chi) \leftrightarrow (\phi \,\&\, (\psi \vee \chi)))$ takes the value 0.

4 Let $U = \{u\}$, the set with one element u. Take u as the interpretation of the symbol **0**. As interpretations of the symbols $'$ $+$ $\cdot$ we take the operations given by

$$u' = u, \quad u + u = u, \quad u \cdot u = u$$

respectively. Then we have an interpretation of the formal language whose domain is a one-element set.

5 (a) 'The operation $\cdot$ is commutative.'

3.4 False. Matrix multiplication in M is not commutative. For example $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 2 & 1 \end{pmatrix}$ whereas

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix}.$$

3.5 True. It can be seen from the table for $\cdot$ that this operation is commutative.

3.6 False. We see from the table that $\alpha \cdot \beta = \beta$ whereas $\beta \cdot \alpha = \alpha$. So the operation $\cdot$ is not commutative.

(b) 'For all x, y in the domain, $x \cdot y' = (x \cdot y) + x$.'

3.4 True. For all matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \begin{pmatrix} r & s \\ t & u \end{pmatrix} \in M$,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} r & s \\ t & u \end{pmatrix}' = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} r+1 & s \\ t & u+1 \end{pmatrix}$$

$$= \begin{pmatrix} a & b \\ c & d \end{pmatrix}\left(\begin{pmatrix} r & s \\ t & u \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} r & s \\ t & u \end{pmatrix} + \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

3.5 True. We check that, for all $x, y \in \mathbb{N}^*$, $x \cdot y' = (x \cdot y) + x$, by considering all possible cases.

$x, y \in \mathbb{N}$	$x \cdot y' = x \cdot (y+1) = (x \cdot y) + x$
$x = 0,\ y = \omega$	$x \cdot y' = 0 \cdot \omega' = 0 \cdot \omega = (0 \cdot \omega) + 0 = (x \cdot y) + x$
$x \neq 0,\ y = \omega$	$x \cdot y' = x \cdot \omega' = x \cdot \omega = \omega = \omega + x = (x \cdot \omega) + x = (x \cdot y) + x$
$x = \omega,\ y = 0$	$x \cdot y' = \omega \cdot 0' = \omega = 0 + \omega = (\omega \cdot 0) + \omega = (x \cdot y) + x$
$x = \omega,\ y \neq 0$	$x \cdot y' = \omega \cdot y' = \omega = \omega + \omega = (\omega \cdot y) + \omega = (x \cdot y) + x$

3.6 True. Again, we check all possible cases.

$x, y \in \mathbb{N}$	$x \cdot y' = x \cdot (y+1) = (x \cdot y) + x$
$x = \alpha$	$x \cdot y' = \alpha \cdot y' = \beta = (\alpha \cdot y) + \alpha = (x \cdot y) + x$
$x = \beta$	$x \cdot y' = \beta \cdot y' = \alpha = (\beta \cdot y) + \beta = (x \cdot y) + x$
$x \in \mathbb{N},\ y = \alpha$	$x \cdot y' = x \cdot \alpha' = x \cdot \alpha = \alpha = \alpha + x = (x \cdot \alpha) + x = (x \cdot y) + x$
$x \in \mathbb{N},\ y = \beta$	$x \cdot y' = x \cdot \beta' = x \cdot \beta = \beta = \beta + x = (x \cdot \beta) + x = (x \cdot y) + x$

(c) 'For all x in the domain, $0 + x = x$.'

3.4 True. For all matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M$,

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

3.5 True. We can easily see this from the first row of the table for $+$.

3.6 False. We see from the table for $+$ that, for example, $0 + \alpha = \beta$.

6 (a) 'For all x, y, z in the domain, $((x + y) \cdot z) = (x + (y \cdot z))$.'

(1) True. For all integers x, y, z, it is true that $(x + y) - z = x + (y - z)$.

(2) False. For instance, $(2 + 3) \cdot 7 = 42$, while $2 + (3 \cdot 7) = 23 \neq 42$, so that the statement does not hold for all real numbers x, y, z.

(b) 'For all x, y, z in the domain, $((x + y) \cdot z) = ((x \cdot z) + (y \cdot z))$.'

(1) False. For instance, $(3 + 4) - 2 = 5$, while $(3 - 2) + (4 - 2) = 3 \neq 5$, so that the statement does not hold for all integers x, y, z.

(2) True. For all real numbers x, y, z, it is true that $(x + y)z = xz + yz$.

(c) 'For all x, y in the domain, there is a z in the domain such that $(x + z') = y$.'

(1) True. We require that, given any integers x and y, there is some integer z such that $x + (-z) = y$. We can do this for any given x and y by taking z equal to the integer $x - y$.

(2) False. For instance, take x to be 3 and y to be 0. Whatever real number we take for z, we will have $z^2 \geq 0$, so that $3 + z^2 \geq 3$, which means that there is no z for which $x + z^2 = 0$.

(d) 'For all x in the domain, there is a y in the domain such that $y' = x$ or $(\mathbf{0} \cdot y') = x$.'

(1) True. For any given integer x, there is an integer y such that $-y = x$: we simply take y to be $-x$. Thus for this y the statement '$y' = x$' is true, so that the statement '$y' = x$ or $(\mathbf{0} \cdot y') = x$' is also true.

(2) True. Take any real number x. If $x \geq 0$, take y to be the real number $\sqrt{x}$, which makes the statement '$y' = x$' true, and thus also makes the statement '$y' = x$ or $(\mathbf{0} \cdot y') = x$' true. If $x < 0$, take y to be the real number $\sqrt{-x}$. Then $(-1) \cdot y^2 = -(\sqrt{-x})^2 = -(-x) = x$, so that the statement '$(\mathbf{0} \cdot y') = x$' is true and thus the statement '$y' = x$ or $(\mathbf{0} \cdot y') = x$' is also true.

7 Let ϕ denote the formula $\forall x\, x = \mathbf{0}$, let ψ denote the formula $\exists x\, x = \mathbf{0}$ and let χ denote the formula $\exists x\, (x = \mathbf{0} \leftrightarrow \forall x\, x = \mathbf{0})$. Then we must show that $(\phi \vee \neg\psi)$ is a tautological consequence of $\neg\phi$ and $\neg(\chi \rightarrow \psi)$; that is, we must show that

$$((\neg\phi \,\&\, \neg(\chi \rightarrow \psi)) \rightarrow (\phi \vee \neg\psi))$$

is a tautology. We consider the truth table.

$((\neg$	ϕ	$\&$	$\neg$	$(\chi$	$\rightarrow$	$\psi))$	$\rightarrow$	$(\phi$	$\vee$	$\neg$	$\psi))$
0	1	0	0	1	1	1	1	1	1	0	1
0	1	0	0	0	1	1	1	1	1	0	1
0	1	0	1	1	0	0	1	1	1	1	0
0	1	0	0	0	1	0	1	1	1	1	0
1	0	0	0	1	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1	0	0	0	1
1	0	1	1	1	0	0	1	0	1	1	0
1	0	0	0	0	1	0	1	0	1	1	0
							↑				
							tautology				

Thus the required tautological consequence is established.

INDEX

$\neg$ 7, 27
$\&$ 7, 26
$\vee$ 7, 28
$\rightarrow$ 7, 28
$\leftrightarrow$ 7, 29
$\forall$ 7, 34
$\exists$ 7, 34
$=$ 7, 34
( ) 7, 34
x_i 7, 34
0 7, 35
$'$ 7, 35
$+$ 7, 35
$\cdot$ 7, 35

$\mathbb{N}^*$ 36
$\mathbb{N}^{**}$ 37
$\mathscr{N}$ 35
$\mathscr{N}^*$ 36
$\mathscr{N}^{**}$ 37
$\beta(\sigma)$ 23
$\gamma(\phi)$ 23
0 27
1 27

addition symbol 7
antecedent 28
arithmetical symbols 7, 35
    interpretation 35
    standard interpretation 35
a-symbol 12, 20
atomic formula 17

basic symbols 7
bi-implication 7, 29
    interpretation 29
brackets 8

conjunction 7, 26
    interpretation 26
connectives 7, 25
    interpretation 25
consequence
    logical 39
    tautological 39
consequent 28

disjunction 7, 28
    intepretation 28
domain 34

existential quantifier 7, 34
    interpretation 34

formal language 4
formal proof 4
formal system 4
formula 17, 19
    atomic 17

Gödel numbers 23

Hilbert's Question 5

identity symbol 7, 34
    interpretation 34
implication 7, 28
    interpretation 28
interpretation 34
    non-standard 35
    standard 35

Leibniz's Question 5, 35
logical connectives 7, 25
    interpretation 25
logical consequence 39
logical symbols 7
logical truth 35
logically valid 39

multiplication symbol 7

negation 7, 27
    interpretation 27
non-standard interpretation 35

principal connective 20
principal symbol 12
propositional logic 33
punctuation symbols 7, 34

quantifiers 7

semantics 4
standard interpretation 35
string 4
subformula 21, 31
    in truth table 31
substring 10
successor symbol 7
syntax 4

tautological consequence 39
tautology 33
term 9, 10
terminal string 16, 20
truth table 26
truth value 26

universal quantifier 7, 34
    interpretation 34

variables 7, 34
    intepretation 34
vel 28

zero symbol 7